Addis Ababa University
School of Graduate Studies
Faculty of Informatics

A FRAMEWORK FOR REQUIREMENTS ELICITATION TECHNIQUES SELECTION

BY

SEMAHEGN ABEBE DEMISSIE

June 2004

A Thesis submitted to the School of Graduates Studies of Addis Ababa University in partial fulfillment of the requirements for the degree of Master of Science in Computer Science

A Framework For Requirements Elicitation Techniques Selection

BY

SEMAHEGN ABEBE

Name and Signature of Members of the Examining Board

No.	Name	Role	Signature
1	Dr. Yirsaw Ayalew	Advisor	
2			
3			
4			

ACKNOWLEDGMENTS

I would like to express my gratitude to all those who helped me to complete this thesis. Special thanks are due to my advisor Dr. Yirsaw Ayalew. He monitored my work and took effort in reading and providing me with valuable comments on major outcomes of this thesis work. His overly enthusiasm and integral view on research and his mission for providing only high-quality work and not less, has made a deep impression on me. I owe him lots of gratitude for having me shown this way of research. I am really glad that I have come to get know him in my life.

I also wish to register my profound gratitude to the MicroLink Information Technology College for the workload protection and partially financing my thesis work. The conducive environment that the college creates during my thesis work also deserves lots of appreciations.

The support, encouragement and advices from the department of computer science were meant a lot to the realization of this thesis.

I also thank the Addis Ababa University, Graduate School for partially funding my thesis work.

I equally wish to extend my appreciation and thanks to my friend Taddesse Tarekegn, Fikreyohannes Lemma, Eyob Alemu, Tamerat Tesfaye and others who encouraged me during my work.

I also feel a deep sense of gratitude for my parents, brothers, and sisters who supported me to come to the accomplishment of this thesis work.

I am very grateful for my wife Etsegenet, for her love, encouragement and patience during the thesis work.

ABSTRACT

The success of a system development is greatly dependent on the quality of the requirements. The quality of the requirements, in turn, is highly affected by the type of elicitation techniques that are employed during the requirements elicitation process. On the other hand, the effectiveness of elicitation techniques is dependent on the situation in which they are used. Some techniques may be more suitable to one situation than others. Little research is done to match elicitation techniques to situations where they are most effective. In addition, the effort to study the driving factors for the selection of elicitation techniques is very small. Most studies focus on advocating specific elicitation techniques or methodologies, instead of providing a guideline on when to use the techniques. This results in serious challenge on analysts, causing them to be confused, being left in the middle of several techniques without the necessary guide on when to use them. This thesis presents a framework, which will assist analysts in the selection process of elicitation techniques that best fit the goal of elicitation session, the project environment and the problem domain. The framework, proposed in this thesis, has three steps and is probe-based. It takes the goal of an elicitation session, information about the project environment and the problem domain and generates list of suitable elicitation techniques in their priority of applicability to a given condition. The result of this thesis work is also included in the ETS prototype, which is a Web based application used to demonstrate the applicability of the framework.

Table of Content

List of Tables

List of Figures

Acronyms and abbreviations

AMA	American Medical Association
ASP	Active Server Page
CDM	Clinical Data Management
ER model	Entity Relationship model
ETS	Elicitation Technique Selection
IBIS	Issue-Based Information System
IIS	Internet Information Services
JAD	Joint Application Development
LEL	Language Extended Lexicon
RAD	Rapid Application Development
RE	Requirements Engineering
SSM	Soft System Methodology
TAD	Total Applicability to Domain
TAG	Total Applicability to Goal
TAPE	Total Applicability to Project Environment
TAS	Total Applicability to Session
UML	Unified Modeling Language

1. Introduction

1.1. Motivation

Most software engineers agree on the significant impact of the requirements engineering process on the success of software projects. Errors that are introduced during the requirements engineering process and are detected late in the testing and maintenance phases costs much more than any other error introduced in other phases. Studies also show that most software errors are introduced during the requirements engineering process [22]. Therefore, improving this process will considerably increase developers' productivity and quality of the software products.

Requirements elicitation is the primary activity in the process of requirements engineering. It is on the basis of elicited requirements that the other activities like specification and validation will be performed. Requirements elicitation often deals with knowledge that is hidden and buried in the depth of the social and organizational structure of the client organization. Thus, the identification of those requirements needs systematic, careful and efficient approaches that do not adversely affect the healthy environment of stakeholders. Hence, to uncover those requirements, we need a proper tool that takes prevailing conditions into consideration.

There are plenty of elicitation techniques and methodologies that can serve as a tool for capturing requirements. Each of those techniques has its own weakness and strength when applied to elicit specific requirements in a given situation. On the other hand, methodologies prescribe list of techniques, which are problem independent solutions. Studies show that the strength of those techniques is complementary. Thus, the basic

question is which techniques to use in a given situation. Wrong tool, at a wrong situation, for a wrong problem results in wrong outcome. Hence, for the success of a project the selection of the right technique that best fits the existing situation is extremely important. For attaining this important goal we need a guiding framework and a software tool that assists analysts to select the right kind of technique at the right time, and in the right situation.

1.2. Research problem

As indicated in the motivation section, ill-defined requirements may result in defective product or even failure of the project. Eliciting requirements from users is a big challenge to most analysts. Some analysts do not know the available elicitation techniques and even those who know, except few experts, do not know when and in what situations they should use those techniques. Hence, there is a wider gap between highly experienced analysts and novice analysts. Expert analysts develop the required skill through their experience and so they do have much success records. Since there is no formal framework guide on when and how to apply requirements elicitation techniques, less experienced analysts face serious challenges. For eliciting user requirements there are lots of techniques and tools but there is 'no silver bullet' that could be applied to any problem in any situation. Therefore, the major problem that this thesis work addresses is the selection of requirements elicitation techniques by developing a framework that will help analysts and managers to select requirements elicitation technique that best suits specific type of requirement, in a given project environment and problem domain.

1.3. Objectives

The general objective of this research is to develop a framework for the selection of the appropriate elicitation techniques that considers selection factors that are derived from the requirement type, project environment, and the problem domain constraints.

The following are the specific objectives:

1. Study the characteristics of the existing elicitation techniques so that it would be possible to classify and organize them according to their characteristics
2. Study the project environment that affect the selection of requirements elicitation techniques
3. Study the impact of domain knowledge on the selection of suitable elicitation techniques. To demonstrate the validity of the selection framework, the health-care domain is chosen
4. Developing a framework that is used to select the right elicitation technique for the right situation
5. Development of a prototype to demonstrate the effectiveness of the framework

1.4. Approach

The overall research activity is shown in *Figure* 1 below. The selection factor is any factor that affects the selection process and is derived from the project environment or type of knowledge to be elicited (i.e., the goal of an elicitation session). The domain constraint is

any constraining condition that comes from the nature of the domain to which the target system belongs.

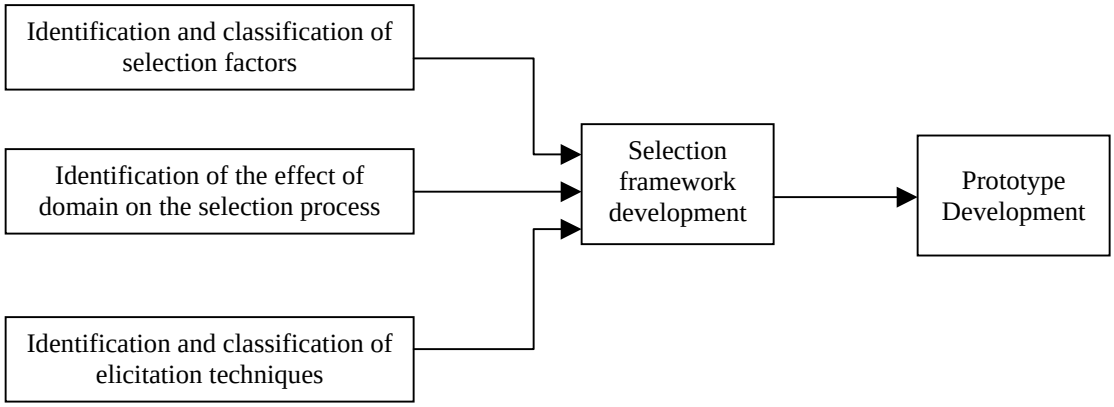


Figure 1: Overall research activities

There are two directions for developing the framework for the selection process. The first one is techniques-driven, that is, studying requirements elicitation techniques and then identifying their suitable application situation. The second approach is problem-driven, that is, to study the project environment and requirements type and search for the appropriate set of techniques. Since techniques are effective in some situations, in following the technique-driven approach, there is a high probability for some situations to stay uncovered. Hence, in this case, the techniques considered in the study will limit the situation's study.

On the other hand, the problem-driven approach needs long time investigation, intensive experience, and exposure to the area of different problems and project environment. Moreover, mapping situations to available techniques may also miss some of the techniques.

In both approaches, there is a risk of incompleteness, but the situation coverage of the two approaches is complementary. Therefore, the combination of the two approaches (i.e., to

study the concepts related with the different elicitation techniques and their nature on the one hand and to study situational conditions for each of the elicitation techniques on the other hand) is selected in this thesis.

1.5. Overview of the thesis

The remainder of the thesis presents what is done with regard to the thesis work. Chapter 2 presents related work while chapter 3 deals with the methodology used in this research work. Chapter 4 deals with identification and categorization of factors from the project environment, the problem domain and requirements type. It also deals with the analysis and categorization of requirements elicitation techniques. Chapter 5 presents the Elicitation Techniques Selection (ETS) framework using a running example (case of clinical data management system). Chapter 6 discusses the ETS prototype.

Finally, in the conclusions and future work chapter, the highlights of the results of the thesis work, its limitations and opportunities for future work are included.

2. Related works

During the past 20 years, software has conquered an essential and critical role in our society [16]. Software applications are crossing the boundaries of almost all disciplines and are serving as effective and efficient tools. But in spite of their growing applications in different areas, the failure record is calling the attentions of researchers and practitioners [44]. Most projects are delayed and run out of budget and even when delivered on time and on budget, projects fail to meet users' need [45]. When asked about the causes of such failure, executive managers identified poor requirements as the major sources of problems [28]. Improving the quality of requirements is thus crucial.

Researchers and practitioners have realized that developing software is not just a matter of creating effective programming languages and tools [16]. They agree on that, the quality of a software product heavily depends on different aspects that have direct or indirect links with the development process such as people, organization, and procedures used to create and deliver it.

Software systems Requirements Engineering (RE) is the process of discovering the purpose for which a software system is developed [37]. Identifying stakeholders and their needs, and documenting these in a form that is amenable to analysis, communication, and subsequent implementation is part of the task that will be done during RE. Requirements engineering can be decomposed into the activities of requirements elicitation, specification, and validation.

As indicated in [4], most of the requirements techniques and tools concentrate on requirements specification, i.e., the representation of the requirements. This thesis work focuses, instead on elicitation concerns; those problems with requirements engineering that

are not adequately addressed. Lots of studies are made on specific elicitation techniques [25]. However, none have yet been done to model elicitation in general. None of the studies provide the foundation that helps analysts to select elicitation techniques that best suite the prevailing conditions of their projects.

The elicitation of requirements is perhaps the activity most often regarded as the first step in the RE process [37]. If right techniques are not used during the requirements elicitation process, some of the requirements of the user may not be identified until late in the implementation and testing phases and that would duplicate the cost many times. The success of large software engineering projects depends critically on the specification, which is a product of information collected during the requirements elicitation process. The specification must represent the requirements of a large number of people with widely differing perspectives [13].

Hickey and Davis [25] present a mathematical model of the requirements elicitation process that shows the role of knowledge in its performance. One meta process of requirements elicitation, selection of an appropriate elicitation technique, is also captured in the model. In their report, they state that the elicitation technique selection is driven by problem, solution, and project domain characteristics as well as the state of the requirements ("The 'right' technique to apply in a given situation must be a function of what requirements we already know and what requirements we still need to know; different techniques are good at uncovering different kinds of requirements"). Their model, which is shown in *Figure* 2, is just a skeleton, it doesn't deal with the technique on how to combine the different inputs of the selection process and get the right set of elicitations techniques.

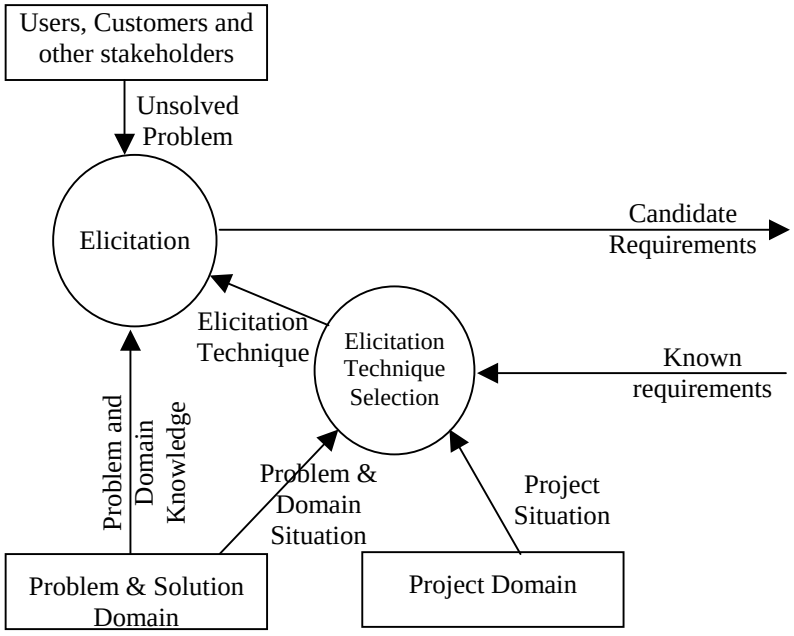


Figure 2: Details of elicitation activities [25]

Nuseibeh and Easterbrook [37] categorize the requirements elicitation techniques into six categories namely, traditional techniques, group elicitation techniques, prototyping, model-driven techniques, cognitive techniques, and contextual techniques. They also suggest that the choice of elicitation techniques depends on the time and resources available to the requirements engineer, and on the kind of information that needs to be elicited. They did not identify the situational conditions on the basis of which the selection of the requirements elicitation technique will be made. They put few techniques as an example in each of the categories. They also didn't exhaust elicitation techniques in each category. Concerning the guidance for the usage of the different elicitation techniques available, they suggested a method-based approach that provides such directions.

Maiden & Rugg [31] presented a framework, which assists requirements engineers to choose methods for requirements acquisition. In their report, they stated that practitioners are often unaware of the range of methods available and even most who are aware do not

foresee the need to use several methods to acquire complete and accurate requirements. They try to bring acquisition methods from software engineering, knowledge engineering and the social sciences together into a single framework. In their study, they identified six facets, which inform the technique selection process: purpose of the requirements, knowledge types, internal filtering of knowledge, observable phenomena, acquisition context and method interdependencies. They didn't indicate the impact of domain on the selection process. Finally, they devised a practical guide for the selection process and provide the ACRE decision support system to provide advice for requirements engineers. Their study does not treat the situational characters independently; moreover, the techniques selected for the study and the involvement of analysts in the formulation process was limited. Their study focused on techniques rather than the project situation.

Hickey and Davis reported [24] that requirement analysts who have extensive experience (and are considered to be masters of elicitation by most) seem to have the ability to select appropriate elicitation techniques on a regular basis. But most practicing analysts have less experience and are more journeyman than master; it is why the software industry fails to satisfy users' need. Thus, Hickey and Davis in their paper suggested that if the average analyst's ability to select elicitation techniques were improved, it would most likely be possible to improve our record of successful software products. Finally, they suggested that mechanisms should be created to formulate the experts experiences so that less experienced analysts could use it easily because less experienced analysts do not know how and when to apply elicitation techniques [24].

As it is reported from different researches, there is a consensus on the importance of a guideline on how to select elicitation techniques that best fit a given project situation. Thus,

the major work in this research is to study the nature of the available elicitation techniques, the situational conditions that affect the selection of elicitation techniques such as factors from project environment and the type of requirements to be elicited, and to develop a framework that assists the selection of the appropriate technique. *Figure* 3 below shows the process of elicitation techniques selection:

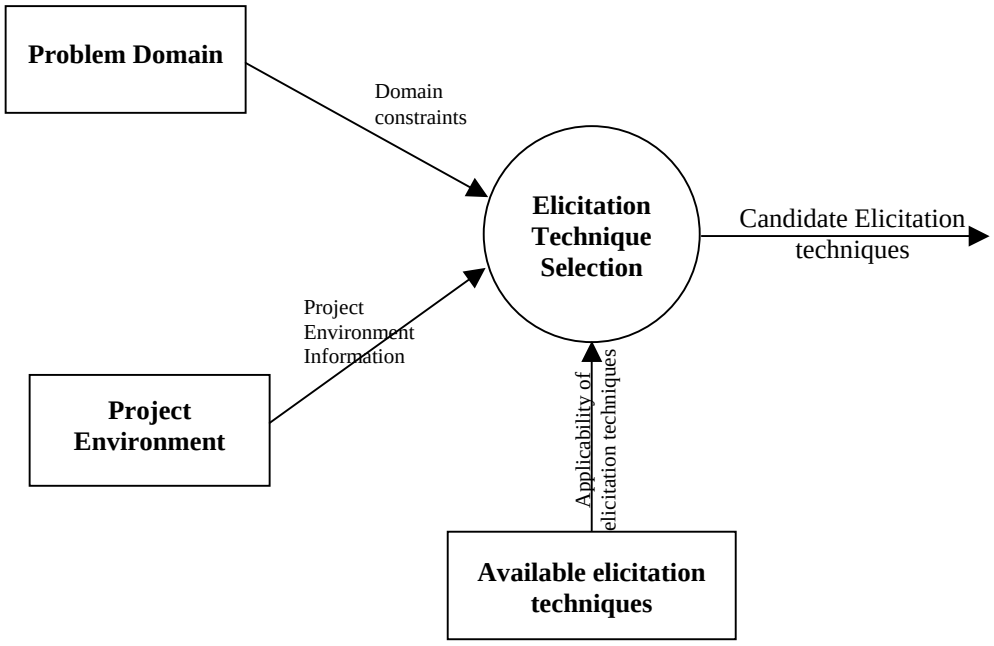


Figure 3: Elicitation technique selection Process

3. Methodology

Qualitative research is characterized by an emphasis on describing, understanding, and explaining complex phenomena - on studying, for example, the relationships, patterns and configurations among factors; or the context in which activities occur. The focus is on understanding the full multi-dimensional, dynamic picture of the subject of study [32].

Quantitative research methods usually involve large randomized samples, more application of statistical inference, and few applications of cases demonstrating findings. In quantitative research information is gathered using predominantly quantitative variables. The objective of quantitative research is to determine the relationship between one thing (an independent variable) and another (a dependent or outcome variable) in a population.

In this research, there are lots of categorization, relationship study, context study, and conditions descriptions. Thus, because of this reason and availability of little information on the topic, inadequacy and missing of relevant theory base for the research, qualitative research methodology is chosen [29].

As indicated in the introduction part, there are five major activities to come up to the final outcome. These activities involve a study and analysis of selection factors, elicitation techniques, domain constraints, framework development and prototype.

3.1. Selection factors

The selection factors are studied in two categories based on the sources of the factors, which are project environment related and requirements type related. Project environment encompasses all factors that come from the context in which the system is developed and

deployed. On the other hand, requirements type deals with factors that come from the kind of requirement that is going to be elicited.

First, all the factors that affect the selection of elicitation technique are listed out from both the project environment and requirements type. Then factors from the project environments are categorized based on their source and their proximity to the project. Factors from the requirements type are categorized based on phases of requirements elicitation process (i.e., starting from collection of context information to conflict resolution and development of stable requirements).

3.2. Elicitation techniques

Elicitation techniques from software requirements engineering, knowledge engineering, cognitive science and social science areas that could potentially be applied for requirements elicitation are studied. A total of 31 elicitation techniques were considered and out of which 26 techniques are selected based on some criteria. The first criterion is the usability of the technique for elicitation of knowledge for a software system or its context (i.e., environment in which the software is situated). Here, the reason for the context knowledge is that requirements of target software systems get their true meaning from their context. The second criterion is availability of documents on the characteristics of a technique for eliciting software requirements.

3.3. Domain constraints

The impact of domain knowledge on the selection of elicitation techniques is clear but it is difficult to get usable information about specific domains or an overview of the nature of domains in relation with elicitation techniques. But there are few studies made on the

health-care domain. Hence, for the study, health-care domain is chosen as a sample representative. The reasons for the selection of this domain are availability of documents in the health-care domain area and visibility of the nature of the domain, as we are all receivers of services from the domain.

The nature of the health-care domain and type of applications that are in use in the domain are studied. The type of elicitation techniques that can efficiently elicit knowledge in the domain and those techniques that are not appropriate to the domain are also investigated.

3.4. Framework development

After having the factors from project environment, requirements type and domain constraints, and the results from the study of elicitation techniques, study is made on how to map the different situational parameters into the set of elicitation techniques. To do that two options are considered, which are either to collect all information concerning the project environments, domain constraints and goal of elicitation session and moving to the elicitation techniques or iteratively apply filtering criteria on the set of elicitation techniques, to get the set of elicitation techniques that can be applied in the given conditions. The second option is chosen because it would be easier for application. The analyst will be provided with probes, batch by batch, which will help to filter the elicitation techniques that are applicable to a given situation. The framework uses three steps to come up with the last set of applicable techniques.

3.5. Prototype development

The prototype is a Web application, which is designed to show how it is possible to automate the selection framework. The major reason for the choice of web application is to

collect comments about the prototype. The prototype will be made available on the net and anyone who is interested to use it to select suitable elicitation techniques would have the access. This would help the validation of the framework with real world problems. Moreover, it will also bring inputs to the later development of the ETS tool.

4. Elicitation techniques and selection parameters

Capturing requirements from stakeholders demands the use of the right kind of elicitation techniques. Studies indicate that there are plenty of available techniques, however, consensus exists among researchers and parishioners that no one elicitation technique can be effective in all situations [24]. A technique could be effective in one situation, incomplete in some other situations and even not applicable in other situations. The challenge of the requirements engineering process is therefore answering the question when to use which technique [19]. For complete, unambiguous, verifiable, consistent, modifiable, traceable, and usable requirements, the positive role of using the right kind of technique is unquestionable. Choice of suitable questioning techniques is not optional rather it is a necessity in any discipline that involves extraction of information from people [41]. In this respect, experienced analysts face relatively less challenge to reach at the right technique that best fits existing situation than novice ones. The challenge is more serious on novice analysts. They don't know which technique to use in what situation [24].

On the other hand, methodologies define a set of techniques to be applied orderly to elicit requirements. When prescribing a technique at a certain stage, they took some assumptions to be true for granted. But the assumptions may not be true in the specific situation in which a methodology is under use. In the presence of a dynamic situation, how could such an assumption possibly be made a priori [25]. Therefore, the question is: how far do these orderly prescribed techniques fit the situation of the problem at hand? Methodologies must also be tailored to the specific problem at hand [39]. Hence, using methodologies is not a solution to the selection of the right elicitation technique. Hickey and Davis [25] suggested a guide for tailoring methodologies to fit a given situation. The guide states that while

applying a methodology, instead of blindly using the prescribed technique, take the technique along with other candidates; study the prevailing situation, then use it only if it is best fit.

Thus, the selection of the right elicitation technique involves a consideration of the situational information. In the next subsections, these situations will be dealt in detail and categorized into groups.

4.1. Selection parameters

The selection of the right elicitation technique is a result of clear understanding of the situation in which the technique is going to be applied. This necessitates the need to study the kind of potential situations in which the elicitation process would be conducted. The sources of factors that affect the selection of the elicitation techniques are categorized into three categories: *project environment*, *requirement type* and *problem domain*.

The **project environment** includes the clients system, the developers system, the nature of the problem, project constraints and other individuals and stakeholders that influence the project. In general, the project environment encompasses stakeholders, the problem and others that can influence the project.

The **requirements type** focuses on the type of information that is going to be elicited at a given point in a project's lifetime. Hence, it can be considered as the goal of the elicitation session. What is the purpose of an elicitation session?

- Is the goal to extract the system context?
- Is it to discover the project mission and goal?
- Is it to clarify ambiguous requirements?

- What is the goal for an elicitation session?

The requirements type deals with the reason why a given elicitation session is going to be held.

The **problem domain** deals with the domain to which the target system belongs. The nature of the problem domain affects the applicability of a given technique in that specific domain.

In the next subsections, the three sources along with factors that originate from them will be identified and classified.

4.1.1. Project environment

The project environment is one of the major sources of factors that affect the selection of elicitation techniques. Stakeholders involved in a given software project may belong to different communities and have different backgrounds but they all have at least one common interest (i.e., the development of the target system). As it is depicted in *Figure* 4, the project environment is divided into five spaces: *the clients' space*, t*he developers' space*, *the problem space*, *the project space*, and *the external environment*.

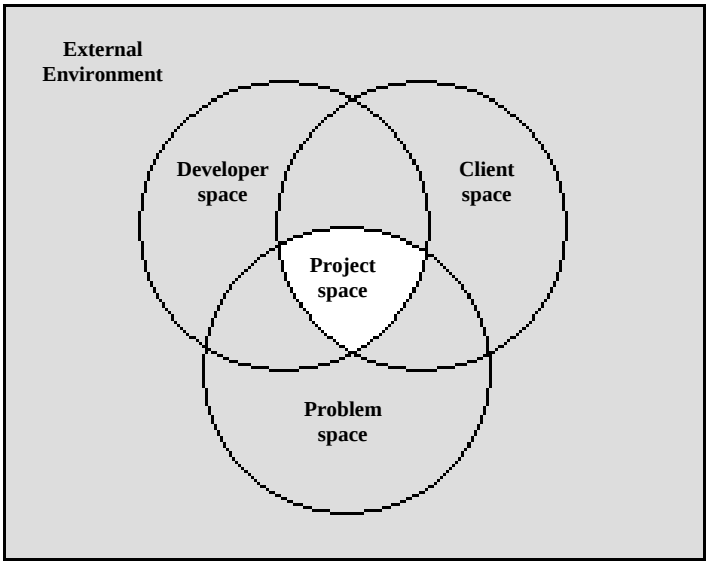


Figure 4: The project environment spaces

Figure 4 shows the four spaces (i.e., the clients', developers', problem and project spaces) as contained in the external *environment* space. This means a factor in the external environment affects all the four spaces. Factors from the external space are not specific to a group of developers, clients, project or even problem domain. It could cross multiple institutions and problem and solution domains.

The client, developer and problem spaces have commonalities with each other as well as among three of them. Developers may be familiar with the problems and solutions from the problem space, which may or may not be useful for the current project. The target project may address some of the problems from the problem space, but this does not mean that the addressed part of the problem space is the whole problem faced by the client. Due to financial or any other constraints the client may want to automate only part of the problem shared from the problem space. Similarly, developers and clients could have lots of interactions and out of the interactions the case of the project can be only part. Therefore, it is the project space that all the four spaces intersect.

The **client space** includes the end users of the target system, who have direct contact with the new system and paying customers. The client space also includes the structure and organization of the users and the customers' enterprises[1]. The clients' space is a bigger space and only part of it will be involved in the development process. In the ideal case, the part outside the involved part shouldn't influence the project, which is because, if it influences, it has to be made part of the project. But, in practice, there are lots of influences on the project from the client system that is not formally included in the project. The political, social and technical realities that reside in the clients' system have significant impact on the selection of an elicitation technique.

The **developers' space** includes the analysts and all those who will be involved in the implementation, testing, quality control, system deployment, and maintenance activities, along with the organization to which they belong. The team structures, the communication mechanism among the teams, the policies and principles of the developers' organization affects the selection of elicitation techniques. In this case too, there are two parts of the space: the part that is included in the project (internal) and the other part, which is outside (external).

The **problem space** includes all conditions related with the problem to be solved and the possible solution area. It can be the complexity of the problem domain, availability of reusable assets in the problem area, rules set on other similar applications.

The **project space** is where the developers and clients interact on the issues of the problem to be solved. All conditions specific to the problem at hand are included in the project space. As indicated above, the clients' space, the developers' space and the problem spaces

[1] Sometimes the users and customers may belong to the same institution (i.e., it is possible for the users' company to be a paying company)

are wider spaces, and they intersect at the project space. The project space is limited by the scope of the target system. How large it includes the other spaces depends on the nature of the problem and other boundary issues.

The **external environment** includes all sources of constraints that are outside the problem, developers' and clients' spaces. It is a source of conditions and constraints that are imposed from government, legal and standard institutions.

Based on the five spaces of the project environment, factors from the project environment are grouped into two: *Internal factors* and *external factors*. Internal factors are those factors that are originated form the project space. On the other hand, external factors are, those factors that are derived from outside the project space. The categorization is made with reference to the boundary of the project space. The hierarchical structure of factors from the project environment is shown in *Figure 5*.

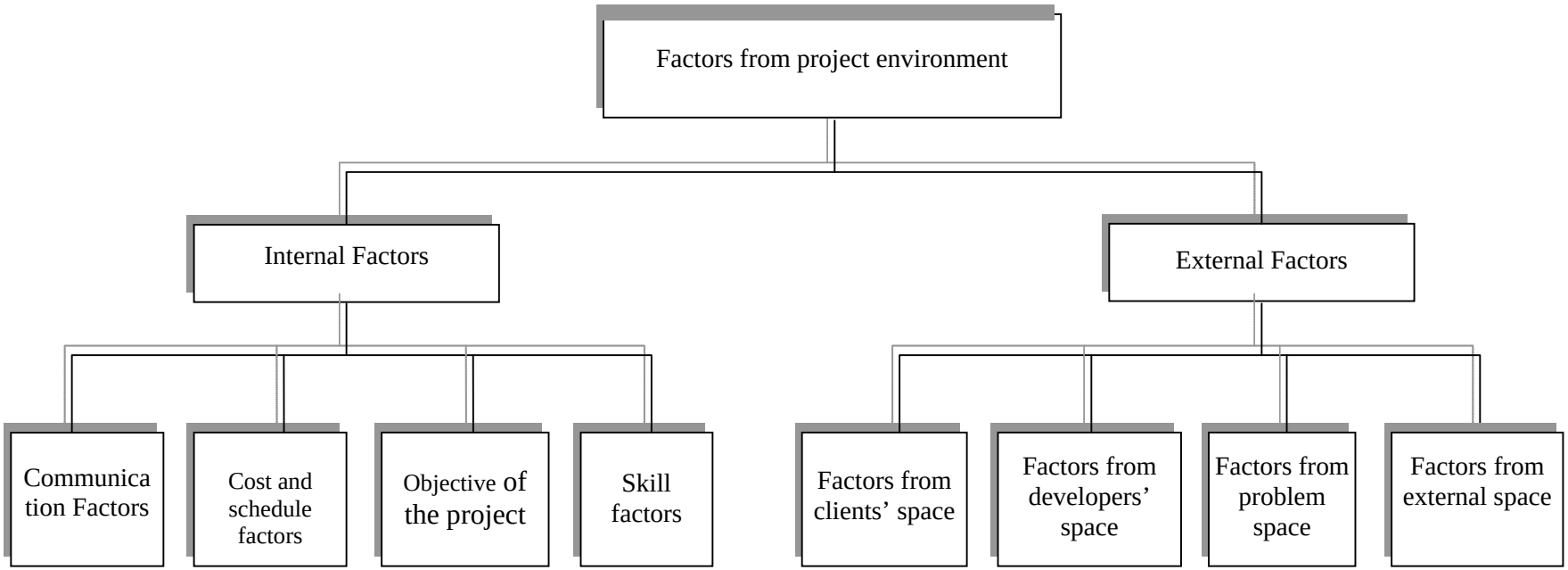


Figure 5: Hierarchical structure of factors form project environment

Internal factors are specific to the target project under consideration. In contrast, external factors are those factors that are not dependent on specific project. External factors include, the structure of the clients' and the developers' organizations, the nature of the problem domain, and other influences of standards, rules and regulations. In the next section the detail of factors under each of the two categories will be discussed.

4.1.1.1. Internal Factors

Internal factors are time, space and people specific. Factors under this category are dependent on the specific project. The maximum lifetime of these factors is the project duration and their influence is bounded by the project boundary. Factors in this category are grouped into five subcategories, which are: *communication factors, cost and schedule factors, objective of the project, and skill factors.*

Communication factors are factors that originate from communication between members of the developers' and the clients' spaces and among members of each space.

Cost and schedule factors are related to the budget allocated to the project and timetable for different activities.

Objective of the project involves all factors that deal with the purpose of the project. The objective can be to develop a new system or to buy off-the-shelf product, or to extend an existing system.

Skill factors deals with skills and experiences of clients, analysts, and developers. This can include familiarity of users to computer technology, skill and experience of developers in a given development approach.

The following section will discuss each one of those subcategories in some detail; factors in each of the subgroups will also be discussed:

A. Communication factors

Requirements elicitation is the most communication intense process [1, 42]. It involves social communicative issues as well as technical issues [47, 48]. The Savant Institute study found that "56% of errors in installed systems were due to poor communication between users and analyst in defining requirements and that these types of errors were the most expensive to correct using up to 80% of available staff time" [22]. Therefore, communication is one of the major factors that affect, not only the elicitation process but also the project success.

Stakeholders (developers, customers, users and analysts) involved in the requirements elicitation process often have different backgrounds and their own interest, which makes their communication complex. The requirements engineer must bring them together to communicate their interest, rectify any conflicting interests and seek for possible solutions. Something, which is common to some group of stakeholders, may be completely new to the other [4].

Factors in communication include lack of the communication skills, inconvenience of bringing stakeholders together because of factors like geographical separations and background differences and exposure to the language used for communication. Some groups may be very familiar with formal languages and models and others may not even have any experience of requirements elicitation process. Therefore, the elicitation process must be adjusted to maximize sound communication among stakeholders. The selection of the right tool that comforts participants should be selected.

The best communication facility also should be selected that minimizes any kind of misunderstandings and ambiguities. For example, in an environment where users are not

familiar with models using model as a communication technique may result in misunderstanding among the analysts and users. Chosen techniques should always favor sound communication among stakeholders. If users have little computer knowledge then it is better to select elicitation techniques that centers users themselves, which will make them part of the development process so that they will not resist the new system. Some elicitation techniques require a higher level of communication skills. For example, JAD require a highly skilled facilitator to enhance communication among stakeholders and to resolute conflicts.

Since users and customers often have difficulty of understanding formal languages, using elicitation techniques that produces natural language could be better. On the other hand, using those techniques may result in ambiguous requirements, which opens the door for multiple interpretations. Thus, the selection of the technique should be dictated by the skills and experiences of the stakeholders. If stakeholders are trained with modeling techniques then it may be better to use modeling techniques because they are good to state requirements that avoids ambiguities.

When stakeholders are geographically far apart from each other, the selection of the appropriate elicitation technique depends on availability of the suitable communication technology. On the other hand, even when stakeholders are close to each other, it may be very hard to organize meetings for identification of requirements. Some of the reasons for this could be unavailability of key stakeholders, or employees may not want to discuss on issues of the company explicitly because of some office politics and other personal reasons. For example, using questionnaires to very busy people makes the rate of respondents very low. Using JAD for people who are very busy is also not a good idea because it may be

very hard to schedule meetings. Key stakeholders may be inaccessible in some situations, which may result in constant change on the captured requirements.

While the main objective of requirements elicitation is extracting the interests of stakeholders, sometimes stakeholders may be unable to articulate their needs. In such situations it is good to use non-verbal techniques such as observation, ethnography, prototyping, and document analysis [40].

Generally bad communication is the killer of a project while good communication, on the other hand, is its redeemer. Therefore, analysts should give serious attention to factors related to communication.

B. Objective of the project

The objective of the project involves the type of customer to whom the system is developed, the nature of the system with respect to existing systems, the main subject of the new system, and other factors, which are normally set before the beginning of the project. The objective of the project is very important to decide which technique to use.

- Is it to develop a new system or to upgrade an existing one?
- Is the product for internal use (i.e., within the company) or for external market?
- Is the system generic or specific to some problem?
- Who is the main subject of the target system? Is it the customer or the users or top managers in the users organization?

If the new system is part of an existing system, it will be constrained by the architecture of the existing system [4]. In such situations, elicitation techniques that concentrate on the study of existing systems, like document analysis, are encouraged.

In some systems the requirement for the user interface is very important. This might be because; novice users will use the system. If this kind of situation exists, it is good to use techniques that help to involve users and to deploy some visual models.

C. Cost and schedule factors

The success of a project can be measured with development of a quality system within the budget and schedule. Financial and time constraints are common in most software development projects. Often it is a great challenge for project managers to meet these two constraints. Cost and time constraints seriously affect the selection of the elicitation technique.

In the presence of time constraints, using techniques that took too much time results in failure to meet deadlines. It takes considerable amount of time to setup some techniques, and some techniques take elongated time per elicitation session and other techniques need much time for data analysis. For example, preparing structured interview or questionnaire, analyzing data after implementing observation technique is a time consuming activity.

The cost of elicitation techniques can be measured with usage of special equipments, cost of training, analysis cost of the collected data, elicitation session setup cost and cost of holding each session. In the presence of tight budget, using techniques, which are expensive, foster the project failure.

In general, whenever a technique is selected it is wise to give due attention to the budget and schedule constraints.

D. Skill factors

Some techniques demand special skills from analysts and other stakeholders and some techniques such as interaction and discourse analysis need formal training. For example,

group techniques need a highly skilled facilitator for successful group session outcome. If clients are not computer literate or don't have elicitation experience they may be too ambitious or too reserved. In this case, tools or techniques that show them the capacity of the current technology, such as prototyping, must be deployed.

If the users are computer illiterate it may be important to concentrate on the usability of the system and to use a technique that facilitates the extraction of best fit user interface and simple to learn system. If developers are not clear with the solution of the problem and have uncertainty on the technical feasibility of the requirements, it is good to use techniques like prototyping.

4.1.1.2. External Factors

External factors, as described previously, are those factors that affect the selection of elicitation technique from outside of the project space. It includes the culture, practices and any other impositions from the clients', developers' and the external environment. The factors in this subcategory are not specific to the target project. Every similar project or any project developed by the same group of stakeholders can be affected by these factors. Based on their source, external factors are grouped into the following four subcategories:

- *Factors from developers' space*
- *Factors from clients' space*
- *Factors from problem space, and*
- *Factors from external space*

A. Factors from developers' space

Experience of developers on the solution domain, familiarity of analysts to the problem domain, organizational policy of the development company influences the selection of an elicitation technique. For example, if analysts are not familiar with the problem domain, questionnaire or structured interview may not suite the project environment. The position of the developers' team in the system development process also affects the technique selection process. For example, in some projects, the developers may be users and customers of the system, which may lead us to the use of introspection technique.

B. Factors from clients' space

In some client organizations, practices may not be documented (i.e., they may not have documented work procedures, manuals and policies). In such kind of organizations, practices can only be captured from those who exercise it and are exercising. In some organizations, the social interaction among employees may be very poor while in others unhealthy competitive spirit may exist. In a healthy environment, users may not have a problem to explicitly state their need. On the other hand, in a hostile environment, they may restrain from explicitly describing their needs. Sometimes vision gap may also exist among employees in the client organization, for example, between the top management body and the actual users of the system, which possibly call the intervention of experts to narrow the gap. Past experiences of client organizations might indicate that there is high staff turnover. This makes documentation of requirements rationales crucially important and techniques that facilitate this should be selected. Organizational goals, policies, and structure and work roles of intended end users also affect the selection of elicitation techniques. Maturity of manual systems should also be considered while selecting elicitation techniques.

C. Factors from problem space

The nature of the problem and solution space also affects the selection of elicitation techniques. What kind of system is the target system?

- Is it interactive?
- Data-centric?
- Real-time? Or
- Distributed?

The modeling that is going to be used for the communication among stakeholders is highly affected by the kind of application that is going to be developed [9]. For example, for interactive systems, it is good to use use-cases and scenario; for data-centric applications, E-R model is the best choice.

In some application development areas like embedded systems, stakeholders have very little impact on the software requirements. Since the hardware design is already there the software requirement will highly be dictated by the hardware architecture. Hence, in such application, the hardware architecture can be used as a source to determine the type of software to be developed.

The other major point is availability of reusable domain assets in the area of the target system domain.

D. Factors from external space

Factors from external space include those factors that are derived from outside of the clients', developers' and problem space. This includes imposition of standards, policies, rules and regulations by standard, legal and government institutions. For example, in some

countries, encryption of data beyond some level is not allowed for the sake of national security. When such kinds of constraints are expected in a given project, it is good to contact the domain experts and referring to legal and standard documents. Factors in this subcategory are cross-domain factors (i.e., they are not specific to the domain of the target system).

4.1.2. Requirement type

The other factor that affects the selection of elicitation technique is the type of information that is going to be elicited. Requirements elicitation activity is not a one-time process rather it is an iterative process. At different times of the software development process needs arise for the elicitation of certain types of information. During the early stages of requirements elicitation, analysts may not be familiar with the client's organization, members' categories, culture and politics in the organization. But at later stages they will become familiar with the new environment. Therefore, the technique that we use during the early project stage will be different from those that will be used in later stages. Thus, at the beginning we may need to use techniques that give us the general picture of the organization and after that we can deploy other techniques that will help us to know the details.

Every elicitation session has one or more goals. The goal can be extracting information regarding the clients' organizational context, identification of the scope of the project, identification of the features included in the target system, detail of the features, existing system evaluation, rationale identification and refinement of already captured requirements. Thus, based on these possible goals, type of requirements can be grouped into six

categories namely: c*ontext study, boundary setting, features identification, system evaluation, rationale identification* and *requirements refinement.*

The six goals are not sequentially ordered. At any point, any one of the goals may get activated, and need arise to extract some kind of requirement. Therefore, it is not a necessity to address the goals in some predetermined order. The execution of the goals depends on specific situation of the project. Thus, the goals are situation driven. For example, if the users are the developers then they may not need to study the context, which is used to get the true interpretation of elicited requirements. In the lifetime of the project, if conflict and ambiguities arise, then requirement refinement and rationale identification may be the solution. In the following section, each group of requirement type will be discussed briefly.

A. Context study

Most current practices concentrate on the computer system without much concern for organizational factors. However, it is from the context knowledge that other requirements would get their true meaning [4]. Requirements elicitation should start with the study of the organizational and contextual nature of the target system. Value system of the organization, work structure, basic aspects of social order, basic category system used by the members, division in social groups, typical pattern of work, and how current technology is in use needs to be identified before the detail requirements elicitation activity.

Context study gives the global view of the environment in which the target system resides. This activity guides or structures the subsequent elicitation activities. It is also through the context that it would be possible to avoid ambiguous requirements that lead to multiple interpretations. Identification of political and power relationships, study of organizational

behaviors, identification of mission of the organization, culture of the organization, and etc highly affect the requirements elicitation activity.

To perform context studies, the right elicitation technique has to be selected. Some of the techniques such as questionnaires need detail knowledge of the organization and categories of the target subjects.

B. Boundary setting

System boundary setting is an activity that is performed to put a demarcation between things that should be included in the project and those that shouldn't. Boundary setting activity involves identification of the goal of a system, high-level statement of the target system's mission or role, architectural constraints, and determination of existence of another system.

In boundary setting there are five activities to be done: identification of stakeholders, determination of operational context, determination of the problem context (i.e., define operational models, goals, and mission scenario), identification of similar systems and performing problem context analysis [4].

Elicitation activities, which are too narrow or too broad, may result in requirements which are ambiguous, non-traceable, unnecessary and non verifiable [4]. Therefore, it is very important to have clear scope information, and for that, the appropriate tool should be employed.

C. Features identification

The primary interest of users is to have a system that will positively affect their environment [12] (i.e., the introduction of the new system must bring a difference in their performance and efficiency). The identification of the important parts of the users' practice

involves three activities – identification of the static nature of the system, its dynamic nature and the data requirements. Features identification involves identification of problems seen by the members as most important, detail study of the problems that needs to be addressed, and identification of data objects. If the features identification is incomplete then it will cause high maintenance cost after the system is deployed. Therefore, to have a stable system we must identify, as many of the features as possible in the early stage of the project. Hence, the question is, which technique is most suitable for the features identification.

D. Rationales identification

In the course of software development, requirements document is used as a reference for conflict resolution and clarifying ambiguous requirements. It is also used as a reference to get the original rationales of requirements when there is a feel of uncertainty in the interpretation of requirements. In an unstable organization where there is a high rate of staff turnover, rationales for previous decisions may be lost along with x-staffs.

In the middle of system development some key stakeholders may be missed. This has a potential for the system to slide away from its original objective. The best thing that must be done, in such situations, is extracting or capturing the reasons of each of the requirements during the elicitation process and documenting it properly for the subsequent phases. Having such a document is also one of the major factors that affect the traceablility of the system. Some techniques, such as Issue-Based Information System (IBIS), are good for identification of rationale and selection of the appropriate techniques.

E. Requirements refinement

The need for holding requirements elicitation sessions can be triggered whenever there is a need to uncover some knowledge. Requirements, after being captured, should be refined to detect and remove any kind of redundancy, incompleteness and conflict. When incompleteness or conflict is detected, holding a refinment process is unavoidable. The refinement process can be categorized into three groups: *Clarification of unclear or ambiguous requirement*, *identification of unknown requirements*, and *identification of conflicting requirements*.

A great deal of uncertainty in requirements, a need for early feedback from stakeholders or high risk on the selection of technology should be clarified. Messy, changing, ill-defined problems, which have the capacity to cripple the project, should be settled down. Identification of unknown requirements requires some sort of stimulant to help stakeholders recall forgotten or neglected requirements. Conflicting requirements may also call for modeling techniques to integrate the requirements and see any conflicts. If the conflicts are already detected then techniques that assist conflict resolution should be deployed.

The quality of requirements is measured by completeness, un-ambiguity, verifiability, consistency, modifiability, traceability, and usability of the requirement. Thus, to reach at this goal, the requirements should be refined.

F. System evaluation

The system evaluation can be made on an existing system to see its drawback and to upgrade it or it can be made on a newly introduced system to see the users response to the new system, which will serve as a requirement for the subsequent maintenance activity.

4.1.3. Domain knowledge

In the past, many studies have been carried out to improve the requirements elicitation process and on elicitation techniques, but little is done on specific domains [8]. Most researches were concentrating on generic solutions, which they think, can be used across domains. However, when we come to specific domains, they have their own special nature, which favor some of the available elicitation techniques. Hence, there is a need to take domain considerations for selecting suitable techniques. Techniques, which could not be used in the domain and techniques, which are suitable for the domain, should clearly be identified. The purpose of this section is to deal with the health-care domain. The nature of the domain and techniques appropriate to the domain and those that are not suitable for the domain will also be discussed.

4.1.3.1. The nature of the health-care domain

The first challenge of the health-care domain is a problem of terminologies that are unique to the domain [8, 43]. There are lots of special terms that are new for an outsider such as the analyst. Different terms could be used for the same concepts and the same terms may have different meanings in different areas. This problem, unless treated earlier, causes misunderstanding among stakeholders.

One way of addressing the terminological problem can be preparing a glossary before getting in to the depth of the problem. This approach has two advantages: first it helps to improve understanding among stakeholders, second it will serve as an asset in the domain for future reuse.

Leite and Franco [30] suggested a Language Extended Lexicon (LEL), which will store symbols representing notions and behavioral responses within the area where the notions and responses are used.

In the health-care domain there is a high concern regarding data privacy, confidentiality, and security. For example, the American Medical Association (AMA) suggests the need for utmost effort and care to protect the confidentiality of computerized medical records. Among the AMA's guidelines are requirements for both the patient and physician be advised about the very existence of computerized medical database. In addition, the AMA requires that this information be communicated to the patient and physician before the physician releases the information as necessary to obtaining the patient's fully informed consent to treatment [33].

The current laws on medical record privacy and confidentiality are not uniform throughout the world and often contradictory [33]. The bundle of privacy rights is based upon ethical, common law and statutory requirements-all of which protect one's autonomy or privacy in one's medical information. It is helpful for analysts, working in the area of health-care domain, to understand these rights, which are sometimes termed "information privacy rights".

The physician-patient relationship is confidential. Were that not the case, most patients would be reluctant to divulge information necessary to the diagnosis and treatment of their problems. There must be a mechanism to protect both the health system and information contained within it from unauthorized access misuse, and accidental damage.

Software applications developed in the health-care domain for medical equipments needs utmost care because they involve the lives of many people. In this domain, technique

selection constraint, range from the availability of stakeholders to the restrictions on the use of some elicitation techniques due to legal and ethical issues.

The health-care domain is one of the areas where there are lots of activities that are still done manually (i.e., in most hospitals and other health-care institutions most activities are done manually) [8]. This creates an opportunity for analysts to reengineer activities in those health institutions to come up with the best way of doing things. Identifying the rationales behind the choice of the different options will assist the analyst to reach at the best way. As Davenport [10] indicated, information technology should support the re-engineering of business so that it can achieve the most efficient way of conducting business.

Most professionals in this domain are new to the usage of computers. For example, in UK, although the country adopt the use of computer system as a national policy for professionals involved in assessing elderly people, only 1% of these professionals are currently using computer systems [36]. This figure, for developing countries, in the absence of a single computer in some hospitals, could even be worse. This shows how difficult using computer can be in this domain. The ignorance of physicians to computer skills along with their busyness necessitates usability of health-care applications crucially important. On the other hand, as indicated previously, the physician-patient relationship should be confidential and this imposes a task of data entry on the physician, which will not be a simple job [43]. Hence, health-care systems need to seriously consider the user interface to make physicians and nurses comfortable.

Health-care domain is characterized by having lots of non-functional requirements as indicated above. Safety, reliability and availability are also crucial requirements in the health-care domain.

It seems that very early on, scientists and doctors alike were captivated by the potential computer technology might have in medicine [5]. But the introduction of computer technology is not as would be expected.

4.1.3.2. Techniques for health-care domain

From the plenty of elicitation techniques available, some of them are more suitable for the health-care domain, some can be used with some modification and some are not even applicable to the domain.

Considering the constraints stated in the previous section, elicitation techniques are rated based on their applicability to the domain. The detail of each of the techniques is shown in *Appendix A.* In the table, the applicability of the different techniques to the health-care domain is quantified. The table doesn't contain all techniques; it only contains techniques studied in this thesis[2]. The table can be expanded to include more techniques and practical experiences of analysts. In the table, the value 0 indicates that the technique cannot be applied in the domain, the value 2 means it is highly suitable to the domain, and 1 means the technique can be applied but is not the best option.

4.2. Elicitation techniques

According to Maiden and Rugg [31], the best elicitation technique is the one that is easy to set, run and obtain requirements; the one that needs only one analyst and one other stakeholder; one that require less technological equipments; and finally one that doesn't

[2] You can find them in *appendix B*

antagonize or bore stakeholders and avoid alienating stakeholders. But, the main issue is: *which elicitation technique satisfies all those conditions in any situation?*" The answer for this question is "*none*"; no technique has all the attributes in all the situations. The type of the requirement to be elicited and the environment in which the problem is situated affects the efficiency and effectiveness of an elicitation technique.

Each of the elicitation techniques may be more suitable for one particular situation than another and therefore, part of the requirements engineering challenge is to choose the right one for each moment [8]. The complex nature of requirements, the sensitivity of the clients system, the cost of error correction introduced during requirements elicitation process necessitates the usage of the right technique to overcome as much of these challenges as possible.

The success of an elicitation technique is measured by how much it helps for the identification of the real needs of stakeholders. If we could find an elicitation technique that is less costly, fast enough, and little impact on the client system, then that technique will be the best candidate for the elicitation process.

As indicated above, the effectiveness of an elicitation technique is highly dependent on the environment in which it is situated. For example, group meeting techniques like JAD/RAD are best candidates for having agreed requirements [4], but if it is difficult to arrange meetings then trying to use the technique may be more costly in terms of preparation time and energy. Therefore, before deciding which technique to use, it is good to study the situation in which the technique is going to be used.

Researches are emphasizing that it is impossible to separate the elicitation process form the environment in which the problem resides. As a result of this, they are trying to include

elicitation techniques from other fields such as linguistics, sociology, psychology and anthropology [20]. Most of the software problems are in some social, cognitive and organizational environments, and this makes them difficult to be elicited by the currently popular software elicitation techniques [38]. That is why requirement engineering is stretching to adopt techniques from other disciplines, even if most of the techniques are not well tested in the software industry.

The choice of the elicitation techniques depends on constraints like time, cost, type of knowledge to be elicited and technology requirements. Nuseibeh and Easterbrook [37] categorize elicitation techniques into six categories based on their nature namely, *traditional techniques, Group elicitation techniques, prototyping, model-driven techniques, cognitive techniques,* and *contextual techniques* [37]. It seems that they categorize elicitation techniques based on their origin, their nature with respect to dealing the context of the system under development, the language used for communication and the number of people involved at a given elicitation session. Their categorization doesn't include those techniques that come from linguistics such as discourse analysis and content analysis. It also excludes simulation techniques like war gamming, role-playing, scenario analysis techniques. In each of the categories, they just indicated few representative examples (i.e., they didn't include as many techniques as possible in each of the categories).

However, in this thesis requirements elicitation techniques are classified into *six* categories: *traditional techniques*, *group techniques*, *simulation techniques*, *cognitive techniques*, *content analysis,* and *immersion techniques.* The categorization is made based on the type and depth of information that the techniques can elicit and the situations in which they can best be applied.

Traditional techniques include techniques that are usually used to collect broad spectrum of information [37]. The main characteristic of most conventional approaches is that they assume preexisting categories. They also consider the system as being static throughout the course of the elicitation activity. For this thesis, only four of those techniques namely *interviews*, *questionnaires*, and *document analysis* are discussed.

Group techniques are techniques, which involves interaction of group of individuals. In this category *JAD/RAD, focus group, team building, Issue-Based Information System (IBIS),* and *Deliphi techniques* are discussed.

Simulation techniques are techniques, which are used to get requirements that are difficult to capture and demands to stimulate stakeholders to enhance the requirements extracting process. In this category, techniques like, *scenario analysis, prototyping, model-based techniques, role-playing,* and *war-gaming techniques* are included.

Cognitive techniques focus on the individual perception of the problem and the solution for the problem. In this category, *protocol analysis, card sorting, laddering* and *Introspection* techniques are discussed.

Content analysis techniques are used to analyze the content. The content can be words, phrases, sentences, paragraphs, pictures, symbols, or ideas. There are many textual, video and audio products of other techniques such as interview, observation, video-based technique, scenario, etc. In this category, techniques, which could be used to analyze those products to elicit more requirements, to clarify existing requirements and to resolute conflicts will be discussed. The techniques, which will be presented, involve *mind mapping, video and audio based elicitation,* and *Requirement Reuse.*

Immersion techniques are techniques that demand the direct involvement of the analyst into the users' system and look and compile the moment-by-moment activity of the clients. In this category, e*thnography, interaction analysis, apprenticing, discourse analysis,* and *observation techniques* will be discussed.

Appendix B contains a total of 26 techniques grouped in to their category and discusses each of the techniques briefly.

5. Elicitation Techniques Selection (ETS) Framework

As discussed in *chapter 4*, the requirements elicitation technique selection takes factors from the project environment, requirements type, and problem domain. Elicitation techniques and their category is also discussed. The remaining question is how to relate the factors and the techniques. There are two options for linking the two: the first is mapping the techniques into the factors and the second is the vise versa (i.e., mapping the factors into the techniques). In the Elicitation Techniques Selection Framework (ETS) the second option is chosen. The reason for that is, because it is the way the techniques are chosen in a natural setup. It is after having the problem that analysts think about any kind of technique. The ETS framework first collects data about the situation of the problem from the analyst and then it will use the data to filter the suitable techniques from the available elicitation techniques.

5.1. The ETS framework

The ETS framework is a three-step probe-based framework. In the framework the analyst will be asked a set of questions which are targeted to identify the conditions in which the problem is situated. The responses to the probes will then be used to filter the set of elicitation techniques that are appropriate to the target problem.

In the first step, information about the goal of an elicitation session will be collected, which will be used to filter those techniques that suite the goal. In the second step, information about factors from the project environment will be collected, which will be used to filter those techniques that suite the project environment and applicable to the goal. The third and final step will further filter the set of elicitation techniques that are already selected to be suitable to the goal and project environment, to get the final set of elicitation techniques

that satisfy the domain constraints too. The general procedure of the framework is shown in *Figure* 6.

The responses taken from the first set of probes will be applied on the set of available techniques **ε** to get another set of techniques **δ,** where **δ** $\subseteq$ **ε** and **δ** will contain all techniques, which are applicable to the selected goal. This set will be used as an input to the second step. In the second step, the responses about the project environment will be collected and used to further filter **δ**, to get another set of techniques (say **Ω),** that suite the project environment. Thus, **Ω** is a set of techniques that are appropriate to the goal and project environment, and **Ω** $\subseteq$ **δ.** Finally, in the third step, domain constraints will be collected and used to further filter **Ω** to get the final set of techniques **ϕ**, which is the set of applicable techniques to the elicitation session and **ϕ** $\subseteq$ **Ω**.

The order in which the three steps are done doesn't affect the final output. As a matter of fact, it is even possible to apply all the filtering factors at once. The reason for dividing the filtering process into the three steps is for the sake of manageability of using the framework in this thesis. But in the future as factors interdependency is studied it would be very important to have those steps because it would be possible to consider only techniques which are interlinked in the selection process.

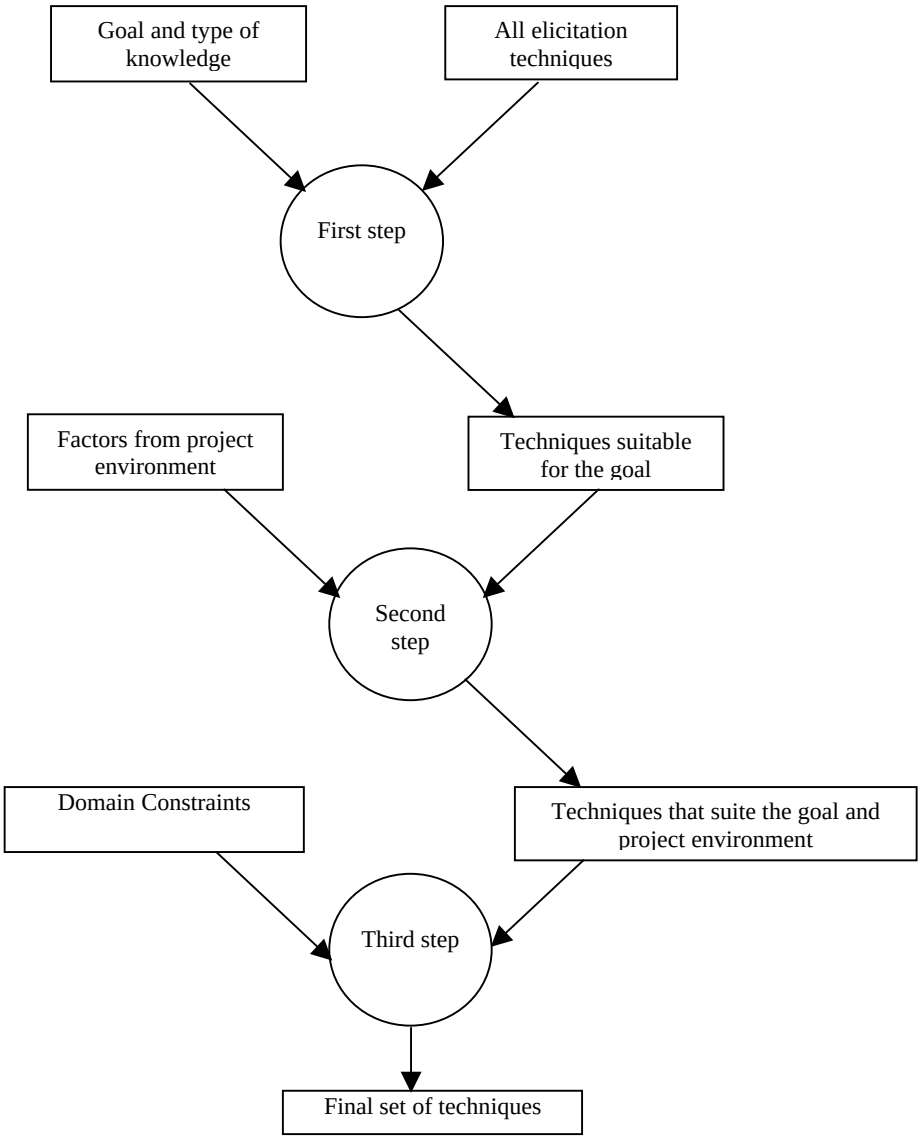


Figure 6: The requirements elicitation techniques selection framework

The detailed discussion of each of the three steps is presented in the following paragraphs.

Step 1: Goal Setting: This first step identifies the goal of the elicitation session. It identifies the reason why the analyst wants to hold the elicitation session. The goal defines the type of the targeted information to be elicited and why the elicitation session is going to be held. The framework presents probes related with the goal and then the responses will be

used to filter those techniques that can be applied to address the goal and elicit the required knowledge. Some of the probes used at this step are:

- Is your goal identification of organizational context?
- Is your goal identification of boundaries of a system?
- Is your goal identification of rationales?

The categories of goals are already discussed in *section 4.2*. Regarding the probes it is possible to have more than one probe for a single category of goal.

In case, the analyst has more than one goal, only techniques that are applicable to all the chosen goals will be selected at this step.

Step 2: Project environment identification: It is not always true that every technique that works for the goal of a given session also works for the project environment. For example, JAD technique is good for the goal of requirements refinement, but if stakeholders are unavailable then, it will be unsuitable for the current elicitation session. This step is designed to identify any kind of factors that are derived from the project, problem, client, developers and external spaces. The responses to the probes, provided by the framework, would be used to further filter those techniques, which are not suitable for the project environment. Hence, the output from this step will be all elicitation techniques that are suitable for the targeted goal and that are applicable to the project environment.

Step 3: Domain identification: In this step the problem domain will be considered. Some of the techniques from step 2 may fit to the goal and the project environment for an elicitation session but they may not be appropriate for the domain of the target system.

Thus, this step is designed to further filter the list of elicitation techniques to get only those techniques that are applicable for the domain under consideration.

Sometimes there is a possibility for a given problem to belong to more than one domain [6]. If the target application belongs to two or more domains, then it is necessary to consider all domains, to which the problem belongs. In the case when more then one domain is selected, the framework gives list of elicitation techniques applicable to all involved domains.

For a given project, the goal of elicitation sessions changes during the course of the project. Each elicitation session targets on the extraction of some information and, after holding the session, we will have another set of requirements that call for another session. Therefore, the goal factor of the selection process may vary for each session unless there is failure of attaining goals, in previous elicitation sessions, or there is a feel of incomplete requirements.

The project environment also changes along the course of the project. The communication among stakeholders may improve, hot political issues may calm down, analysts may become familiar with the techniques, and available domain assets may latter be discovered. Thus, at each of the selection sessions, the framework will track any change on the values of factors from the project environment.

When it comes to the domain constraints, it is the most stable factor. Often domain constraints would have the same value throughout the lifetime of a project. This parameter may only need to be considered when some change in the domain is expected. For a single project, reconsideration of the domain parameter will be important, if the project takes

several years, or the elicitation is for an existing system that was under use for many years without maintenance.

From the above discussion, we can conclude that the goal of elicitation session is the most dynamic and the domain constraint parameter is the most stable while the project environment is somewhere in the middle.

5.1.1. The ETS selection rule

The ETS selection rule shows how a set of elicitation techniques will be selected for a given elicitation session. In this subsection, we will see how we apply responses for the goal, project environment and domain to filter elicitation techniques that fits the condition.

From the studies made on the project environment, requirement type, domain constraints and the study of elicitation techniques, suitability of techniques to given situation is rated, numerically.

The degree of applicability of a technique for a given goal, project environment, or domain is rated with values ranging from 0 to 2. The value 0 indicates that the technique cannot be used for the given condition and the value 2 indicates that it is highly recommended to use the technique for the situation or the technique is an ideal option. The value 1 indicates that the technique can be used in the specified situation but is not the best technique.

For example, using JAD in a situation where the stakeholders are very busy and arranging meeting is hardly possible, is impractical. In the same way, using ethnography in the presence of tight schedule constraints is not applicable; in situations like this, the value 0 is used for the applicability of a technique to the given factor.

Applicability of a questionnaire for requirements refinement can take the value 1. It means that the technique can be used but it is not the most recommended because it has lots of constraints and limitations that could potentially result in wrong or incomplete putputs. When the size of stakeholders is extremely large and there is diversity among users, using questionnaire may be the best option. If the project is under tight schedule-constraint using interview can be an ideal solution. In such conditions, the value 2 is given for the applicability of the technique to the situation under consideration.

Probes that are designed to investigate the goal and other information will be provided to the analyst. Then, based on the responses to the probes, the list of the elicitation techniques that are applicable for the specific problem and project situation will be selected from the available elicitation techniques. The techniques, which will be selected, are only those techniques whose applicability value to all the conditions considered is 1 or 2. Techniques with applicability value 0 for one of the conditions, are totally inapplicable. For a given technique, if it is inapplicable for one of the conditions, then that technique will not be selected at all.

The applicability of the technique to a given elicitation session is called Total Applicability to a Session (TAS). TAS is calculated as the sum of the Total Applicability to Goals (TAG), Total Applicability to the Project Environment (TAPE) and Total Applicability to the Domains (TAD) of the specific technique.

Suppose we have goals g_1, g_2, g_3, …, g_p and techniques t_1, t_2, t_3, …, t_k and the applicability value of the techniques to the goals as shown in *Table* 1. The **TAG** value of a technique $\mathbf{t_i}$ for a given elicitation session will be calculated as follow:

$$\mathbf{TAG_{ti}} = \sum_{r=1}^{p} \mathbf{ag_{ri}}$$

where $\mathbf{ag_{ri}}$ refers to the applicability value of the technique t to goal g_r and $ag_{ri} \neq 0$ for all r = 1..p

Goal	**Techniques**				
	t_1	**t_2**	**t_3**	**…**	**t_k**
g_1	ag_{11}	ag_{12}	ag_{13}	…	ag_{1k}
g_2	ag_{21}	ag_{22}	ag_{23}	…	ag_{2k}
g_3	ag_{31}	ag_{32}	ag_{33}	…	ag_{3k}
…	…	…	…	…	…
g_p	ag_{p1}	ag_{p2}	ag_{p3}	…	ag_{pk}

Table 1: Techniques applicability to goals

Similarly, suppose we have project environment factors $\mathbf{f_1, f_2, f_3, …, f_n}$ and techniques $\mathbf{t_1, t_2, t_3, …, t_k}$. The applicability of each of the techniques to the environment is as shown in *Table* 2. Thus, **TAPE** of a technique $\mathbf{t_i}$ for a given elicitation session will be calculated using the following equation:

$$\mathbf{TAPE_{ti}} = \sum_{k=1}^{n} \mathbf{af_{kid}}$$

where $\mathbf{af_{kid}}$ refers to the applicability value of the technique t_i to factor f_i with degree d, and $af_{kid} \neq 0$ for all k=1..n

The applicability of a technique depends on the option selected for the factor from the project environment. For the degree, we have a range of 5 values from 0 to 4, the meaning of the values to each of the factors, is different. The list of the factors that are considered in this work along with the values to each of the degree is shown in *Appendix C*. The analyst will fill the degree of intensity of that situation, represented by the factor, in the existing

situation. After the choice of the degree, only those techniques whose applicability value to the given factor and the chosen degree is non-zero, will be selected.

Factor	Degree	Techniques				
		t_1	t_2	t_3	…	t_m
f_1	0	af_{110}	af_{120}	af_{130}	…	af_{1k0}
	1	af_{111}	af_{121}	af_{131}	…	af_{1k1}
	2	af_{112}	af_{122}	af_{132}	…	af_{1k2}
	3	af_{113}	af_{123}	af_{133}	…	af_{1k3}
	4	af_{114}	af_{124}	af_{134}	…	af_{1k4}
f_2	0	af_{210}	af_{220}	af_{230}	…	af_{2k0}
	1	af_{21}	af_{222}	af_{233}	…	af_{2k4}
	2	…	…	…	…	…
	3	…	…	…	…	…
	4	af_{214}	af_{224}	af_{234}	…	af_{2k4}
f_3	0	af_{310}	af_{320}	af_{330}	…	af_{3k0}
	1	af_{311}	af_{321}	af_{331}	…	af_{3k1}
	2	…	…	…	…	…
	3	…	…	…	…	…
	4	af_{314}	af_{324}	af_{334}	…	af_{3k4}
…	…	…	…	…	…	…
	…	…	…	…	…	…
	…	…	…	…	…	…
	…	…	…	…	…	…
	…	…	…	…	…	…
f_n	0	af_{n10}	af_{n20}	af_{n30}	…	af_{nk0}
	1	af_{n11}	af_{n21}	af_{n31}	…	af_{nk1}
	2	…	…	…	…	…
	3	…	…	…	…	…
	4	af_{n14}	af_{n24}	af_{n34}	…	af_{nk4}

Table 2: Techniques applicability to project environment

In the same way, suppose we have domains $\mathbf{d_1, d_2, d_3, \ldots, d_m}$, and techniques $\mathbf{t_1, t_2, t_3, \ldots, t_r}$. If the applicability of each of the techniques to the domains is as shown in *Table 3*, then **TAD** for a given technique $\mathbf{t_i}$, can be calculated as:

$$\mathbf{TAD_{ti}} = \sum_{s=1}^{m} \mathbf{ad_{si}}$$

where $\mathbf{ad_{si}}$ refers to the applicability value of the technique t_i to domain d_s and $ad_{si} \neq 0$ for all s = 1..m

Domain	**Techniques**				
	t_1	**t_2**	**t_3**	**…**	**t_k**
d_1	ad_{11}	ad_{12}	ad_{13}	…	ad_{1k}
d_2	ad_{21}	ad_{32}	ad_{23}	…	ad_{2k}
d_3	ad_{31}	ad_{32}	ad_{33}	…	ad_{3k}
…	…	…	…	…	…
…	…	…	…	…	…
d_m	ad_{m1}	ad_{m2}	ad_{m3}	…	ad_{mk}

Table 3: Techniques applicability to domain

Finally, total applicability of technique t_i for a given elicitation session, **TAS**, can be calculated as follows:

$$\mathbf{TAS_{ti} = TAG_{ti} + TAPE_{ti} + TAD_{ti}}$$

For a given elicitation session, having higher value of TAS, means that the technique is the best candidate for the session. After having the techniques in their priority, it is the task of the analyst to judge the technique to be used in the current situation.

5.1.2. ETS application example

In this section, Clinical data management (CDM) system is used to illustrate the three steps of the ETS framework and to see its effectiveness. In some part of the demonstration, screenshots from the ETS prototype are used.

In the sample demonstration, only project environment factors that are common to most projects are considered. Concerning goals, 7 options are used for the demonstration. Most of them are taken from the requirements type discussion in *section 4.1.2*. The goals are adjusted in a way that analysts could easily understand them. **ε,** which is the list of all available techniques, the set of all elicitation techniques discussed in this thesis work are taken.

Step 1: Goal identification

As it is shown in *Figure 7*, suppose, "identification of organizational context", is chosen as a goal for an elicitation session in the CDM system development.

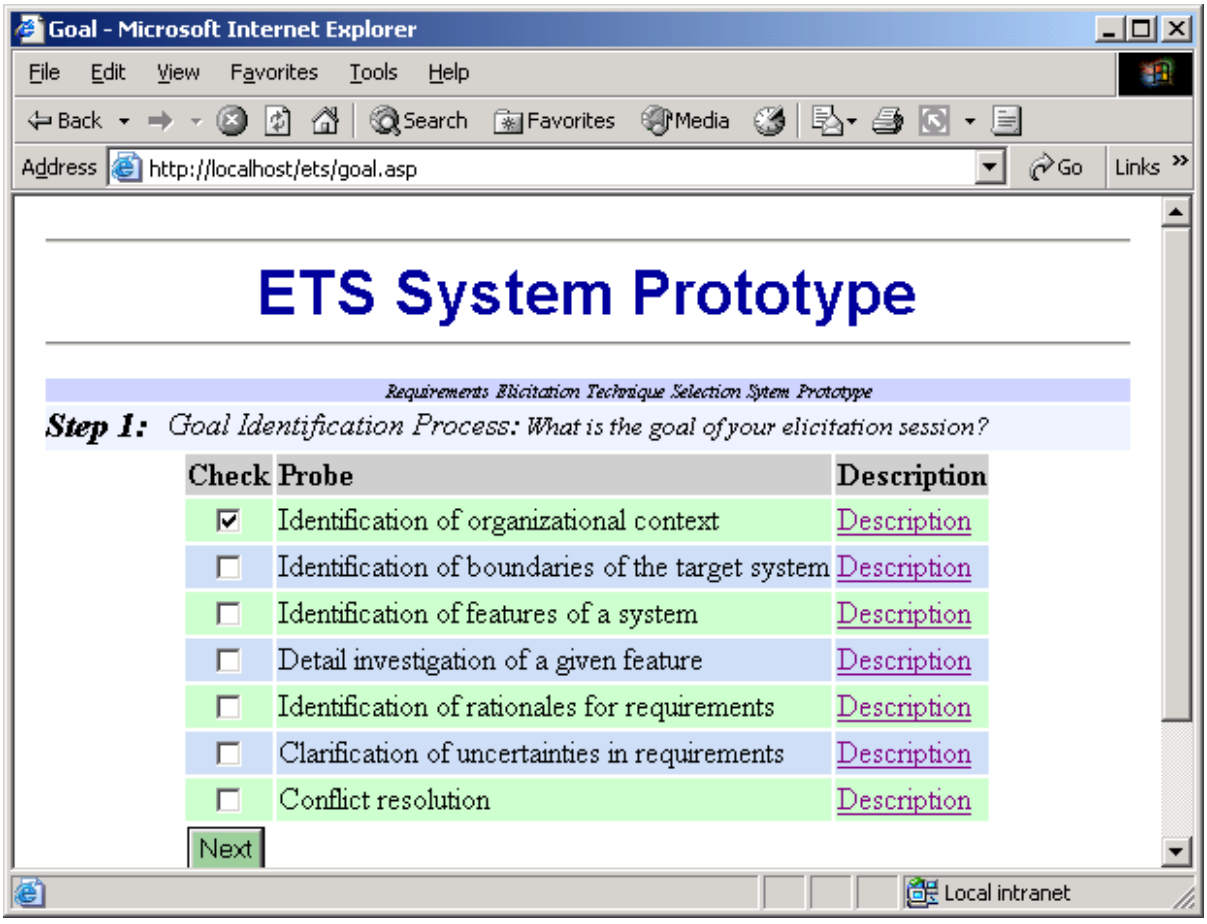


Figure 7: Goal selection window

The following table contains the applicability of each of the techniques to the *"identification of organizational context"* goal. The table is part of the applicability matrix of techniques to goals shown in *Appendix D*.

Techniques																								
Document Analysis	Questionnaires	Structured interview	Unstructured Interview	JAD session	Protocol Analysis	Video and Audio	Scenario Analysis	Observation	Ethnography	Discourse Analysis	Prototyping	Introspection	Card sorting	Laddering	Focus Group	IBIS	Modeling	Apprenticing	Team building	Requirement Reuse	Mind mapping	Delphi Techniques	War gamming	Role playing	Interaction Analysis
2	**1**	**1**	**2**	**1**	**0**	**1**	**0**	**2**	**2**	**1**	**0**	**1**	**0**	**2**	**1**	**1**	**0**	**0**	**1**	**2**	**1**	**0**	**0**	**0**	**1**

Table 4: Techniques applicability for extracting organizational context

Since those techniques whose applicability value is 0 will not be included in the candidates list, the set of the applicable techniques that suite the selected goal are only 17 of the 26 techniques:

δ = {*Document Analysis, Questionnaire, Structured Interview, Unstructured Interview, JAD Session, Video and Audio, Observation, Ethnography, Discourse Analysis, Introspection, Laddering, Focus Group, IBIS, Team Building, Interaction Analysis, Requirements Reuse, Mind Mapping***}**

Step 2: Project environment identification

At this step, the analyst is expected to provide information regarding the project environment of the project. In this section, 16 probes are filtered from the study of the project environment and elicitation techniques that were described in *Chapter 4. Figure 8* below shows the screenshot of the ETS prototype and the selections made regarding the sample problem.

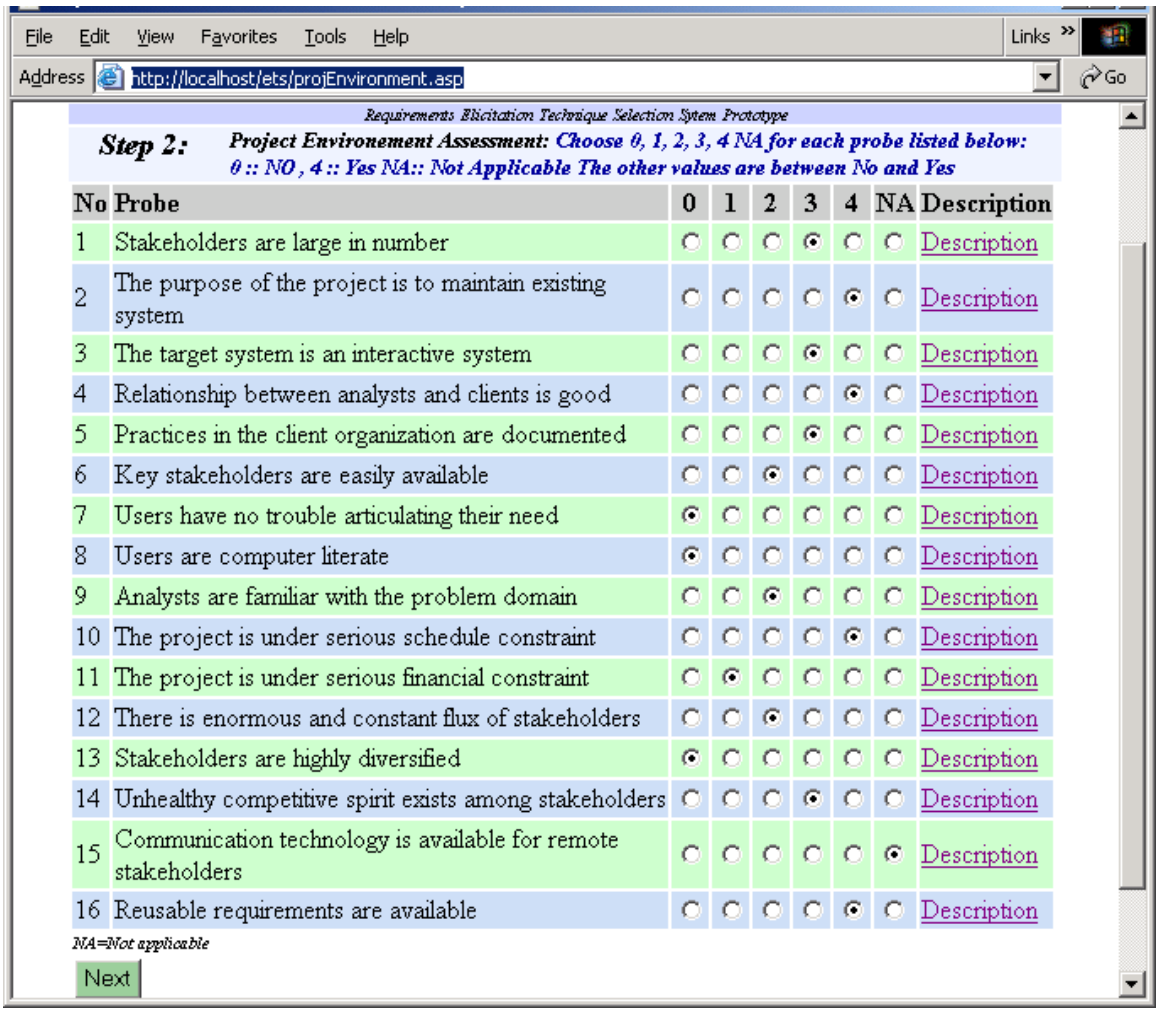


Figure 8: ETS project environment data entry form

In the figure above NA stands for Not Applicable. For example, the probe "Meeting technology is available for remote stakeholders" has a value NA (i.e., the probe is not applicable for the current situation). This may be because all stakeholders are in the same location.

The applicability value[3] for each of the selected options for the techniques that are filtered in step 2 is shown in *Table* 5 below:

[3]The applicability matrix for the different options of each of the project environment factors is listed in *Appendix C*

Probe	Techniques																
	Document Analysis	Questionnaire	Structured Interview	Unstructured interview	JAD session	Video and Audio	Laddering	Introspection	Interaction Analysis	Observation	Discourse Analysis	Ethnography	Focus Group	IBIS	Team Building	Requirements Reuse	Mind mapping
The purpose of the project is to maintain existing system	1	1	1	1	2	1	1	1	0	1	1	1	2	2	1	1	1
The target system is an interactive system	1	1	1	1	1	1	1	1	2	1	1	1	1	1	1	1	1
Relationship between analysts and clients is good	1	1	2	2	2	1	1	1	1	1	1	1	1	2	0	1	1
Practices in the client organization are documented	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Key stakeholders are easily available	1	1	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1
Users have no trouble articulating their need	2	1	0	0	0	2	2	2	1	2	2	2	0	0	1	1	2
Users are computer illiterate	1	1	1	1	2	1	1	1	1	1	1	1	2	2	1	1	1
Analysts are familiar with the problem domain	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
The project is under serious schedule constraint	1	1	1	1	1	0	1	2	1	0	0	0	0	0	0	2	0
The project is under serious financial constraint	1	1	1	1	1	1	1	1	1	1	1	2	1	1	1	1	1
There is enormous and constant flux of stakeholders	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Stakeholders are highly diversified	1	1	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1
Unhealthy competitive spirit exists among stakeholders	2	1	0	0	0	2	1	1	1	2	1	2	0	0	2	1	1
Communication technology is available for remote stakeholders	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Reusable requirements are available	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2
Stakeholders are large in number	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
TAPE	**20**	**17**					**17**	**18**								**18**	

Table 5: Applicability Matrix for the Sample example

As can be seen, only 5 out of the 17 techniques that were applicable to our goal, are applicable to the project environment specified to the framework. The rest of the techniques contain one or more 0 values for the existing environment. Hence, the new set, which is the output of step 2, will contain only 5 techniques as shown below.

ϕ = {*Document Analysis, Questionnaire, Laddering, Introspection, Requirements Reuse***}**

These 5 techniques are applicable to satisfy our goal and the existing project environment considered for the sample problem.

Step 3: Domain identification

In this step, the applicability of the 5 techniques, which we got from step 2 will be checked against the health-care[4] domain constraints.

The applicability value of the 5 techniques to the health-care domain is shown in *Table* below.

Domain	Techniques				
	Questionnaire	Document Analysis	Laddering	Introspection	Requirements Reuse
Health-Care	**0**	**2**	**2**	**1**	**2**
TAD	**0**	**2**	**2**	**1**	**2**

Table 6: Health-care Domain Constraints for the Sample Example

As shown in the table, questionnaire has 0 value for the health-care domain. The reason for that is the low response rate experienced in the domain. As it is explained by Cysneiros [8], although it is a known weakness of questionnaires, low response rate seems to be particularly serious in the health-care domain. One potential reason is related to the fact that quite frequently nurses and physicians work in shifts and

[4] The health-care domain constraints are shown in *Appendix A*

frequently, the initial hours of these shifts are work intensive, trying to catch up all the work that was previously done by the other team. As an experiment is done on three different laboratories and two different hospitals, less than 10% of the distributed questionnaires were handled back and out of them more than 80% was partially filled [8]. Document analysis, requirements reuse and laddering in the health-care domain are suitable techniques because they assist in dealing with the terminological problem in the domain. Introspection can also be used in case analysts are familiar with the domain.

In summary the total applicability of the techniques to the session under consideration is shown in *Table 7* below.

No.	Technique	TAG	TAPE	TAD	TAS
1	Document Analysis	2	20	2	24
2	Requirements Reuse	2	18	2	22
3	Laddering	2	17	2	21
4	Introspection	2	18	1	21

Table 7: List of techniques applicable to the elicitation session

Considering the total applicability value to the session, TAS, document analysis is the best candidate technique for the elicitation session under consideration. The other techniques are also applicable but they can be considered if the document analysis cannot be applied due to the analyst's personal preference or any other reason that is not addressed by the framework. The screenshot of the final output window of the ETS prototype is shown in *Figure 9* below

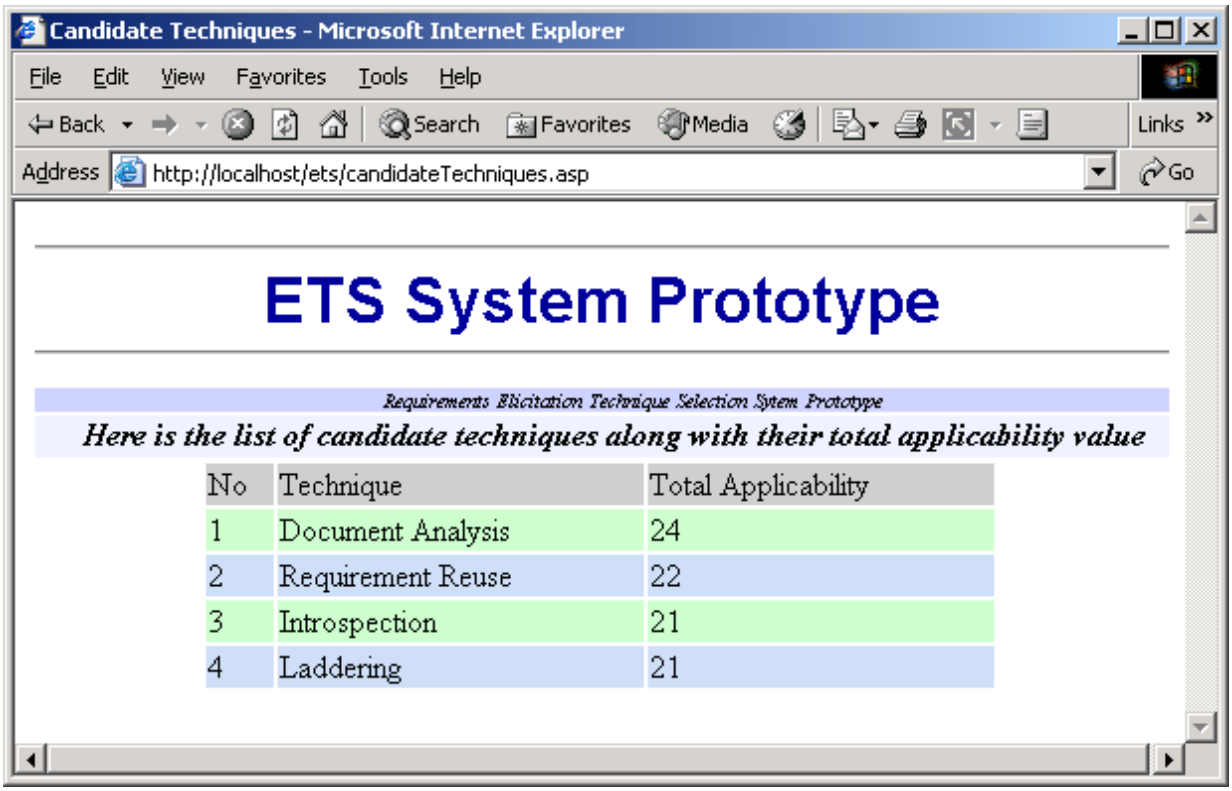


Figure 9: Candidate elicitation techniques displayed by the ETS Prototype

6. The ETS Prototype

The main objective of this prototype is to demonstrate the automation of the requirements elicitation techniques selection process. The prototype also serves as a preliminary work for future development of the ETS tool.

6.1. Scope of the prototype

The ETS system takes selection factors that are identified in the thesis work. It is restricted to the health-care domain. The probes used in the prototype are also limited to the list of probes identified in this research work.

We store all the data, derived from the three areas (i.e., goal, project environment and domain), in a database. The quality and precision of the suggestion of the system is dependent on the quality and number of parameters stored in the database. Thus, from this study it can be concluded that as the database gets richer and richer, the quality and precision of the techniques suggested by the framework will increase. This decreases the workload on the analyst. In this regard, the system will provide interfaces to enrich the available filtering resources, which are stored in the database. The best system is the one that considers as many techniques and situational information but gives precisely few techniques.

6.2. Features of the system

The ETS system prototype provides the following features:

- **Parameters Service**: As it is indicated *section 6.1*, the selection factors are not exhausted (i.e., there is a need to update the factors database). Hence, this service is

used to deal with the factors in the database. It provides interfaces for the addition, deletion and modification of factors and their values.

- **Techniques Service**: The elicitation techniques used in the prototype can be modified, a new technique could be added, or a technique may be deleted using this service of the prototype system. This service will provide its own interface for addition and modification of elicitation techniques.

- **Probe processing service**: The selection process starts by providing probes to the analyst and then the system will process the responses to the probes to select the appropriate set of techniques. The system provides an interface containing the probes. The responses will then be analyzed to select the appropriate technique that fits the situation.

6.3. Development tools

The prototype is a Web-based application. The reason for that is to make the system available on the Web so that analysts could use and evaluate it. This may help in the realization of the future plan, which is the development of the ETS tool.

Active Server Page (ASP) is free with Internet Information Services (IIS), which is free with Windows and most of our laptops and desktops use windows operating system. Hence, ASP is chosen for the prototype development. Moreover, ASP can be written using VBScript, an easy-to-learn scripting language that uses syntax similar to Visual Basic's. Since there is a good chance that a number of people already know Visual Basic, it would be possible for many developers and researchers to understand the ETS prototype easily.

Concerning the database, Microsoft Access and an OLEDB connection is chosen. The reason for this selection is that MS Access is an excellent prototyping tool that may be used to establish the functionality of a database development before upsizing it to interface with existing corporate databases held as Oracle or SQL Server.

6.4. System Architecture

The main design goals for the ETS prototype is to provide systematic database system structure that can be expanded when new factors are identified within the three categories of selection factors.

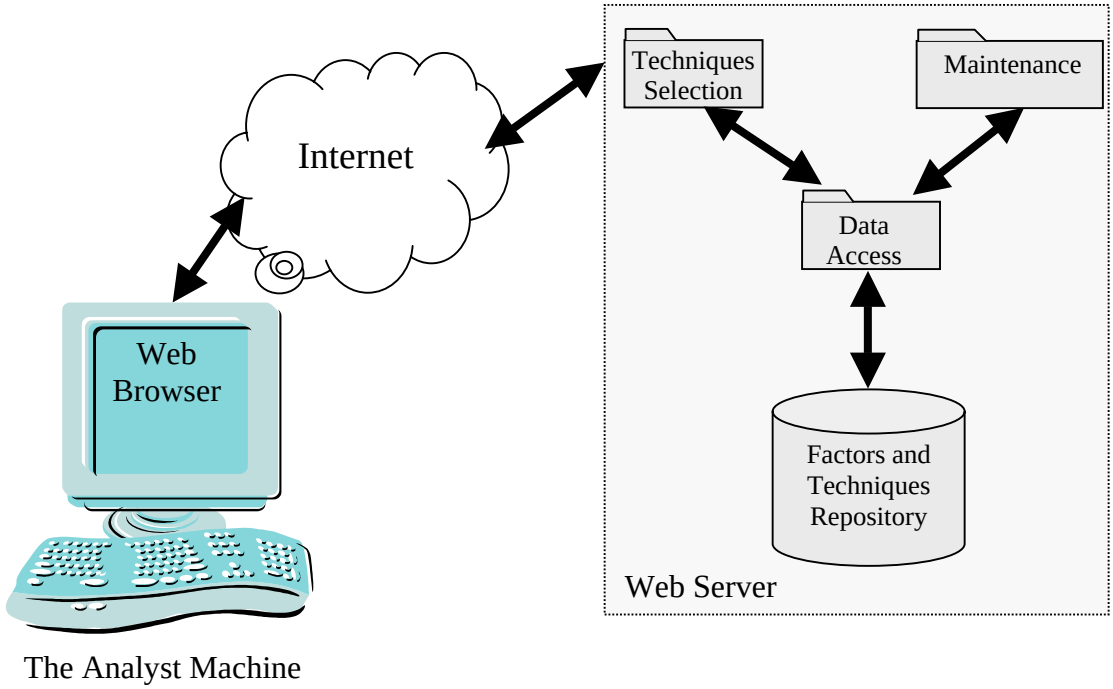


Figure 10: System architecture of the ETS framework

As it is shown in *Figure* 10 there are three subsystems namely- *Maintenance subsystem*, *techniques selection subsystem*, and *Data Access subsystem*. The main objective of the Maintenance subsystem is modification of the system for any expansion to include new factors or to modify existing factors. It is also responsible for the modification of elicitation techniques in the database. The techniques selection subsystem is responsible for providing

list of probes, for use by the analyst to provide the project situation and to generate a set of appropriate elicitation techniques that fit the situation. This subsystem uses the selection rule that is discussed in *chapter 5* to filter the suitable technique. The third subsystem, the data access subsystem serves as an interface between the previous two subsystems and the database. A brief discussion regarding the subsystems is provided below:

1. **Maintenance subsystem**: This subsystem provides addition, modification, and deletion features of the prototype that includes the modification of the selection factors and elicitation techniques from the database. When new factor is identified and there is a need to put it into the repository or when there is some modification to an existing factor in the repository, this subsystem is responsible. The other important point is that, whenever there is any kind of modification of factors, the selection subsystem will check any conflict that may happen to the data in the repository.

2. **Techniques Selection subsystem**: This subsystem is responsible for accepting the analyst's responses from the three sets of probes. The goal setting probes, the project environment identification probes and the domain selection probes. In this implementation of the prototype, multiple domains are not considered; rather only the health-care domain is considered. This subsystem is responsible for providing the probes and passing it to the data access subsystem.

3. **Data Access subsystem**: It has two major purposes. The first is to accept the response of the analyst for the probes from the techniques selection subsystem, and return those techniques that best fit the current condition of the project. The other

task is whenever there is any modification on the factors it will check for any problem of consistency.

6.5. The Class Diagram

There are four class entities, which are *Technique*, *Domain*, *Project_Factor*, and *Goal*. The Techniques class is related with the Goal, Domain, and Project_Factor classes through the applicability of the techniques to the goal, the domain and the project environment. The class diagram shown in *Figure 11*:

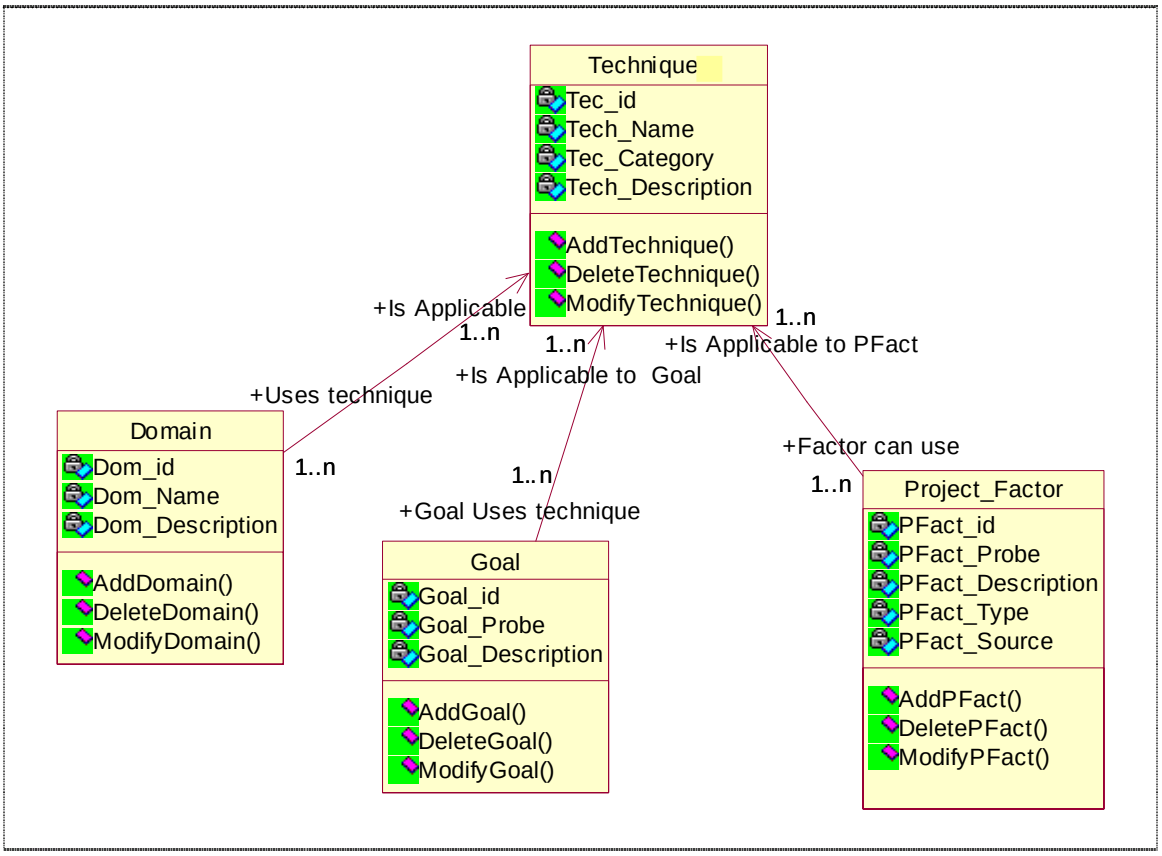
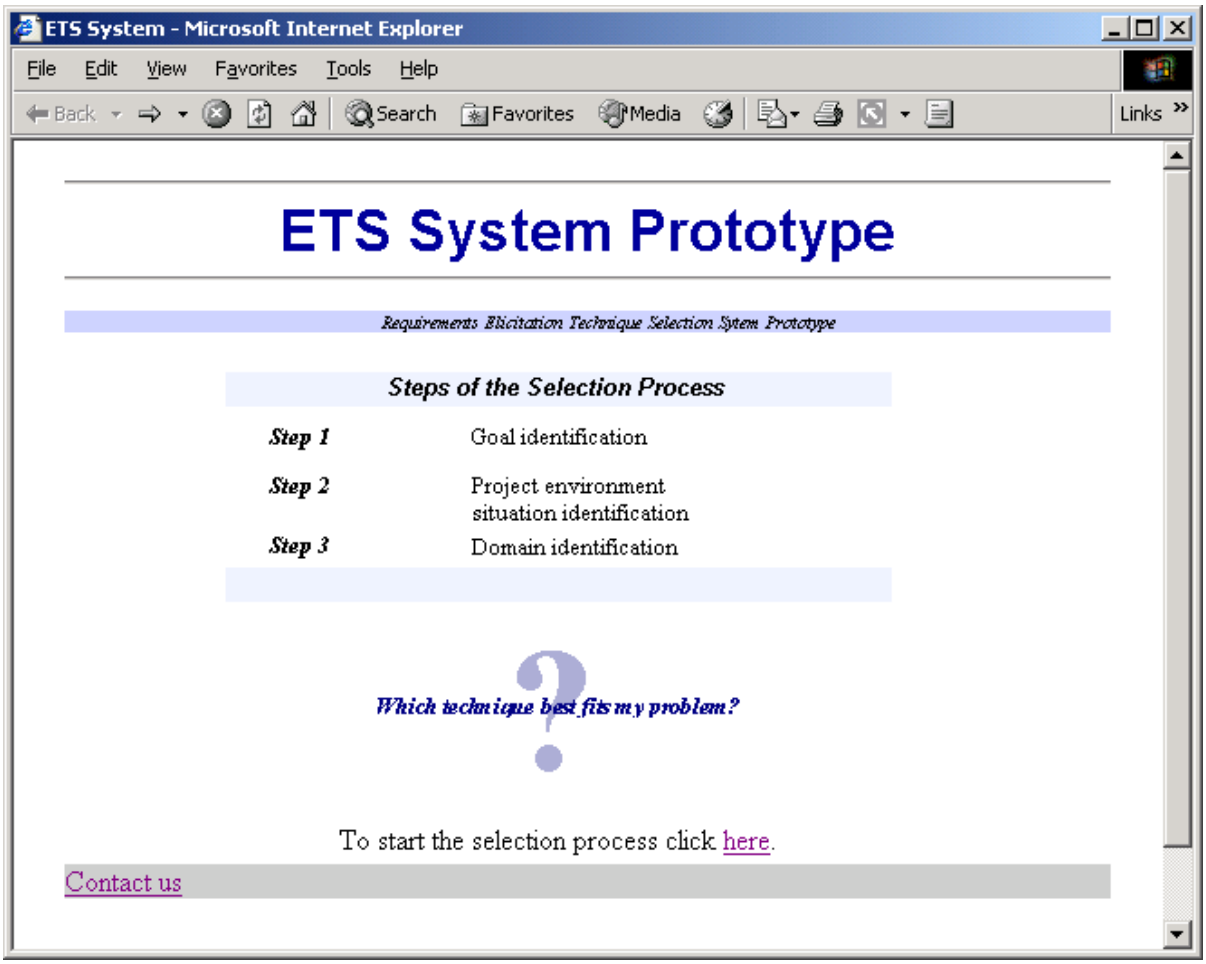


Figure 11: The Class diagram for the ETS system prototype

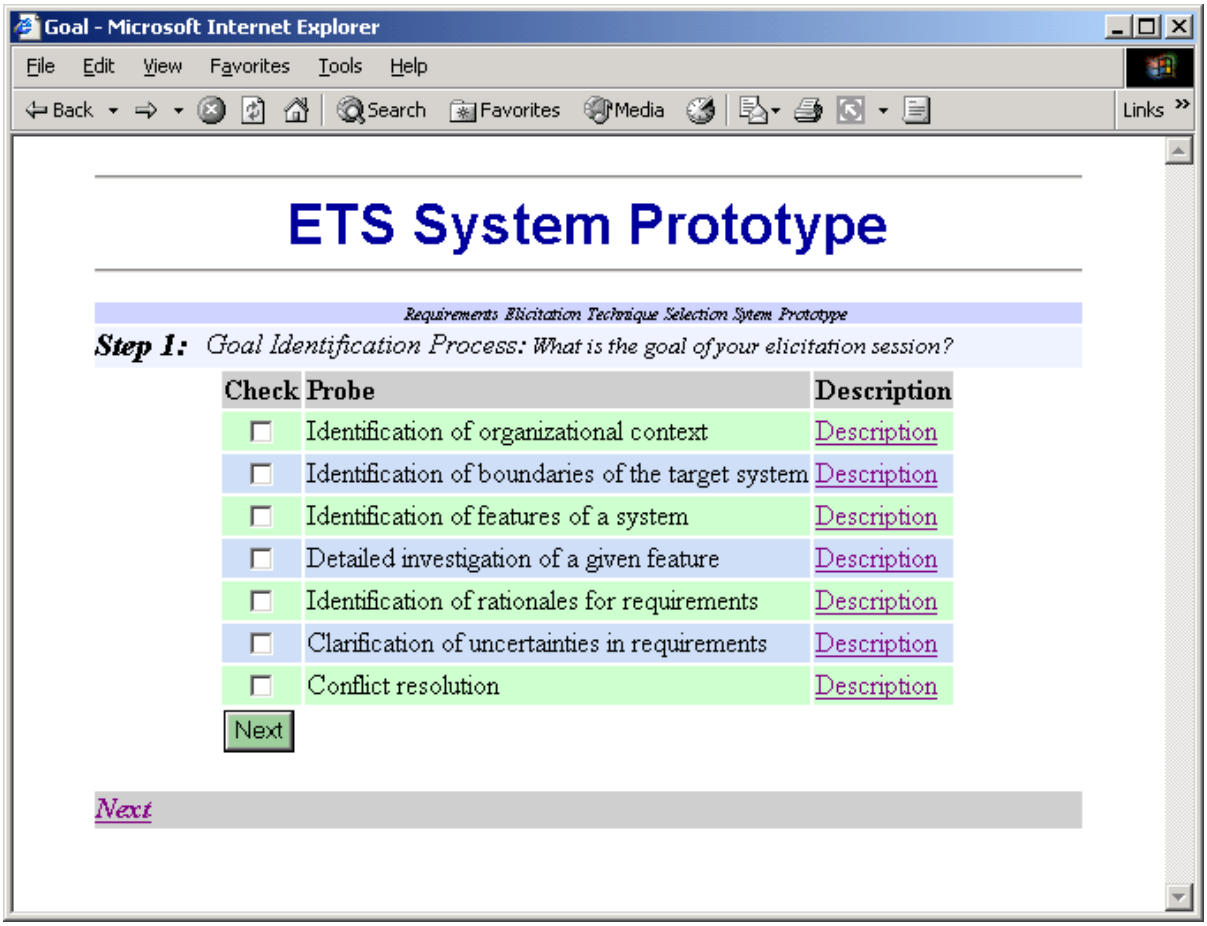
6.6. Interface design

There are five major pages for the system

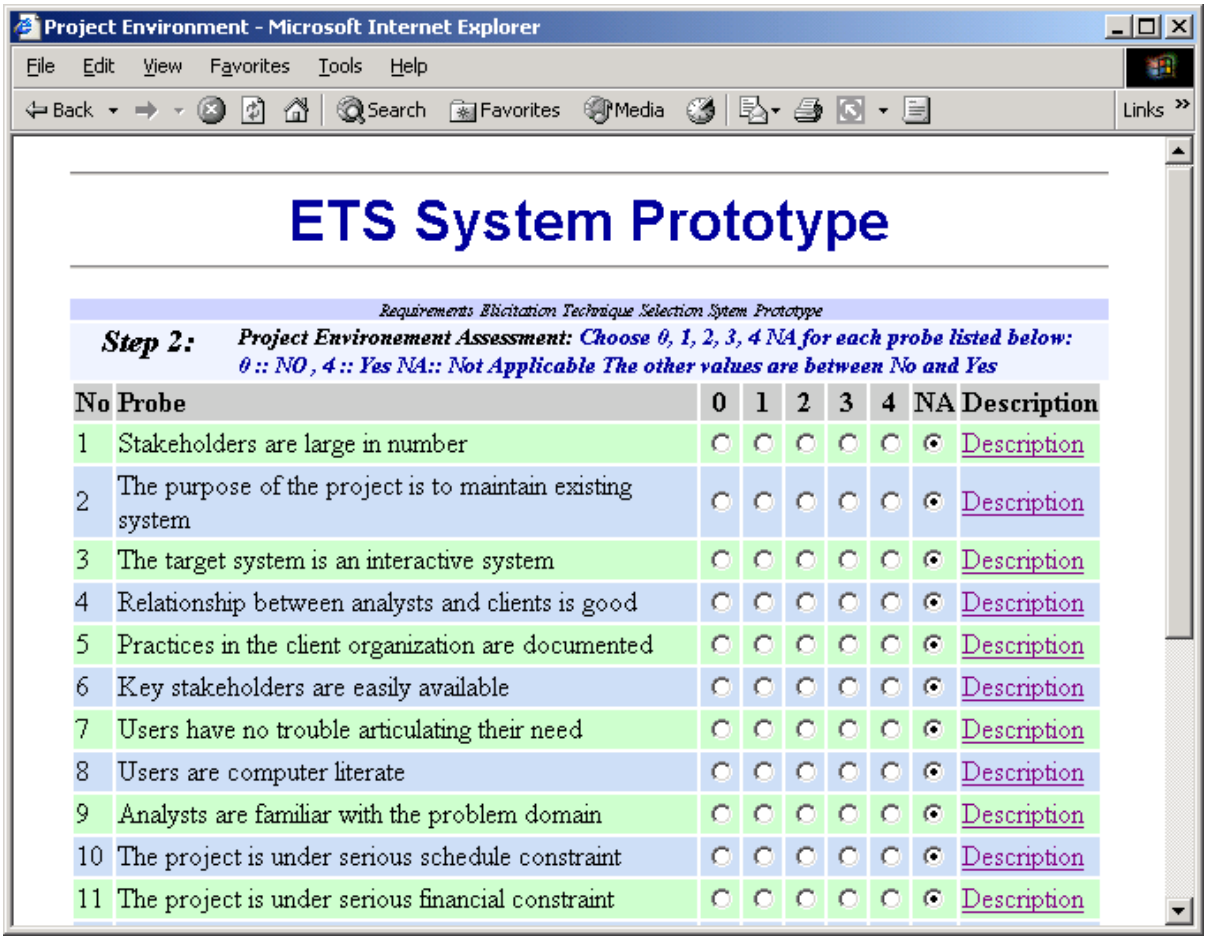
1. The selection home page: Describe the three steps of the ETS framework and contains a link to the first step (i.e., the goal identification). The screenshot of this page is shown below:



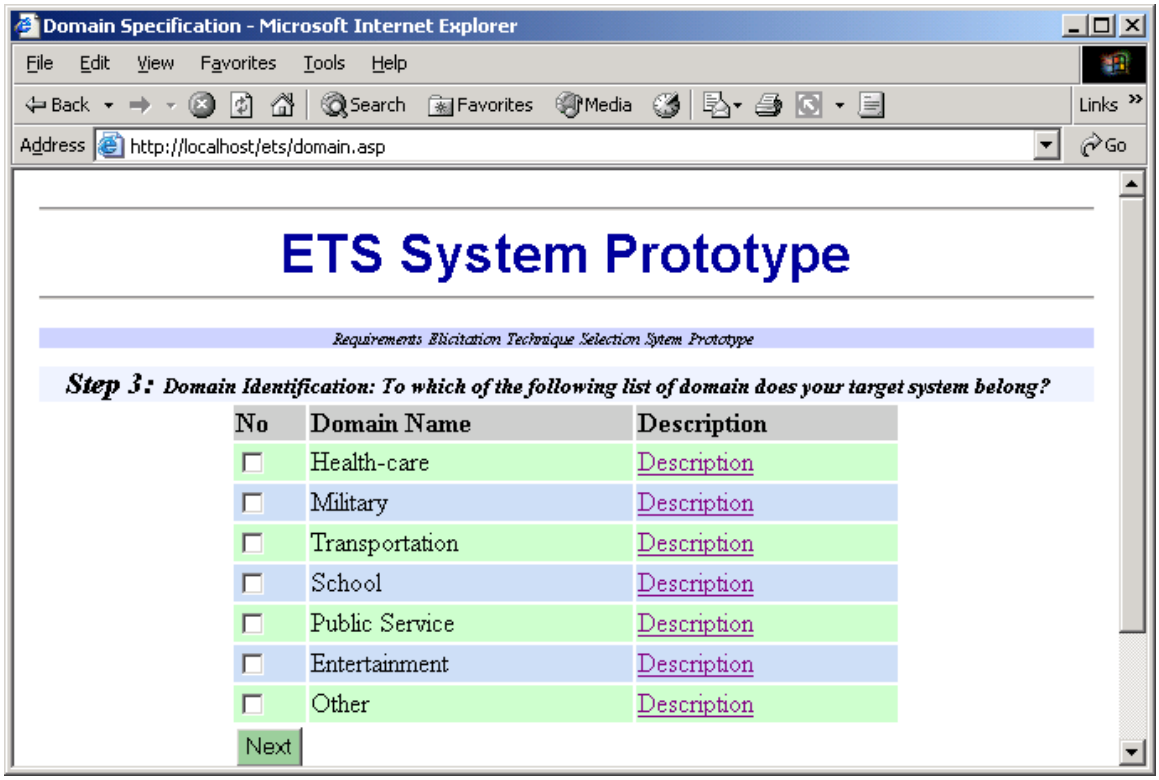
2. Goal setting page: this page is used to probe the analyst for the goal of the elicitation session. A screenshot of this page is shown below:



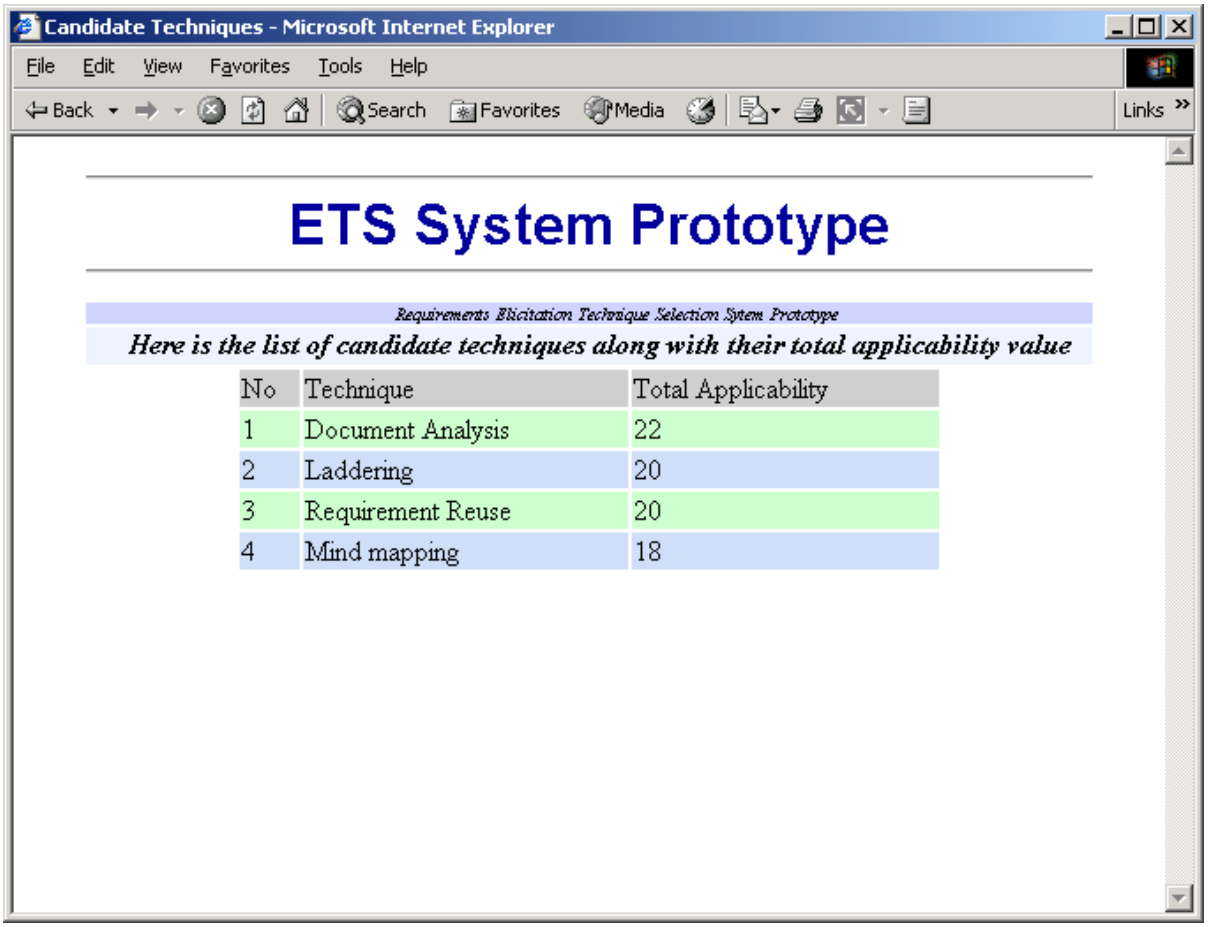
3. Project environment Assessment page: used to receive information about the project environment. A screenshot of this page is shown below:



4. Domain identification page: It is used to track the domain of the target system. A screenshot of this page is shown below:



5. Candidate techniques list page: It displays list of elicitation techniques that can be applied for the problem in the given situation. A screenshot of this page is shown below:



6.7. Discussion

The degree of precision of the result of the ETS prototype is highly dependent on the exhaustiveness of the considered situations and domain constraints. For the sample, only conditions that are common to most projects are considered. But as more conditions are taken into consideration, the result will be more refined.

The depth of the study made on each technique is also an important factor to the precision of the result. As a technique is studied in detail, its appropriateness to a given situation would become more and more clear.

The third factor that affects the precision of the result is the range of techniques that are considered. In this thesis work, to demonstrate the framework, 26 techniques are taken but as the number of techniques that could be taken as an input to the ETS framework increases, more chance will be created to come up with the ideal technique.

The result from the ETS framework is different from similar research. The works by other researchers, as far as the researcher knowledge is concerned, are all qualitative and do not consider all the three factors that are considered here. In this thesis work, the relationship between situations and techniques is quantified. This helps analysts to clearly see best candidates of elicitation techniques for their condition.

7. Conclusions and Future work

Even if the ETS framework needs more refinement and through test, it is believed that it could assist both experienced and novice analysts in the process of elicitation techniques selection. It also helps software project managers during project planning.

The final result of the ETS framework is a set of candidate elicitation techniques that can be successfully utilized for a given requirement type, in the given project environment, and in the presence of constraints from the problem domain. The further selection of an elicitation technique from the candidate techniques can be done using some heuristics.

7.1. Limitations

There are some assumptions and limitations in this research work. It is assumed that the number of factors, which affect the elicitation techniques selection, is manageably finite and the effect of each of the factors is equal. It is also assumed that for any factor there is at least one elicitation technique, which can be successfully applied.

Unavailability of organized knowledge on all domains, which will hardly be realized in the future too, limits this thesis work to only the health-care domain. Hence, as more studies are made on more domains, the effectiveness of the ETS framework would increase.

The number of elicitation techniques taken in this thesis work is limited (i.e., only 26 techniques are dealt in this paper). As more elicitation techniques are considered, the probability of finding a technique that would precisely fit a given situation would be very high.

7.2. Future work

The ETS framework, defined in this thesis, will be the basis for lots of new research directions. A few are introduced below:

1. Assessment of expert analysts' experiences while choosing elicitation techniques for specific situations. In this thesis experience of expert analysts is not included, study of their experience may be important to make it available to novice analysts and managers.

2. In some situations, the final candidate elicitation techniques, generated by the ETS framework may have equal or very close total applicability value. Further study is required on how to make decisions after having the list of such list of techniques.

3. The list of factors for the selection of elicitation techniques needs more refinement and factors interdependence also should be studied.

4. More studies are required on problem domains to see applicability of elicitation techniques to specific domains

5. Design of knowledge-base system to serve as a repository of the elicitation techniques, along with their applicability to specific requirements type, project environment and problem domain constraints.

6. Social, cognitive and organizational issues are at the heart of many of the problems facing RE and because they cannot be addressed solely by the currently available software engineering techniques, novel approaches and paradigms are being sought from other disciplines. Thus, it is very important to have interdisciplinary studies to see any opportunities for improving the requirements elicitation process.

References

[1] R. Alvarez, Discourse Analysis of Requirements and Knowledge Elicitation Interviews, Proceedings of the 35th Annual Hawaii International Conference on System Sciences, 2002.

[2] G. Alistair Sutcliffe, A. M. Maiden, Shailey Minocha, Darrel Manuel, Supporting Scenario-Based Requirements Engineering, IEEE Trans. Software Eng., vol. 24, No. 12, pp.1072-1088, December 1998.

[3] F.P Brook, No Silver Bullet: Essence and Accidents of Software Engineering, Computer Magazine, April 1987.

[4] Christel, Michael G., Kyo K.C., Issues in Requirements Elicitation (CMU/SEI-92-TR-12, ADA258932), Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, September 1992.

[5] E. Ciera, Guide to Health Informatics, second edition, 2003

[6] P. Clements, L. Northrop, Software Product Lines: Practices and Patterns, Addison-Wesley, 2002.

[7] Conklin, Jeff, Begeman, L. Michael gIBIS, A Hypertext Tool for Exploratory Policy Discussion, ACM Transactions on Office Information Systems, pp:303-331, October 1988.

[8] L.M. Cysneiros, Requirements Engineering in Health Care Domain, in Proc. of the IEEE Joint 10th International Requirements Engineering Conference,

the IEEE Joint 10th International Requirements Engineering Conference, September 2002.

[9] K. Czarnecki, Generative Programming: Principles and Techniques of Software Engineering Based on Automated Configuration and Fragment-Based Models, PP:33-66, 1998

[10] T.H. Davenport, J.E. Short, The New Industrial Engineering: Information Technology and Business Process Redesign, Sloan Management Review, PP: 11-27, 1990

[11] O. Dieste, N. Juristo, A.M. Moreno, J. Pazos, A. Sierra, Handbook of software Engineering, vol. 0, No. 0, World Scientific Publishing Company, 2000.

[12] E. Dubois, J. Hagelstein, and A. Rifaut, Formal Requirements Engineering with ERAE, Philips Journal of Research 43(3/4): 393-414, 1988.

[13] S. Easterbrook, "Elicitation of Requirements from Multiple Perspectives, University of London, Department of Computing, Imperial College of Science, Technology and Medicine, June 1991.

[14] European Usability Support Center, Usability Methods, http://www.lboro.ac.uk/eusc/g_interviews.html, September 1998, 4/9/2004

[15] R. Fabian, Domain Engineering, http://www.fabian.ca/shrc/dome.htm, April 2003, 4/15/2004.

[16] A. Fuggetta, Politecnico, Software Process: A Roadmap, Proceeding of the conference on The future of Software Engineering, pp.25-34, 2000.

[17] G. Gaffney, What is card sorting, Information and Design, 2000.

[18] Garner, Rochelle, Why JAD Goes Bad, http://www.computerworld.com/cwi/story/0.1199.NAV47_STO1472.00.html, 1994, 4/9/2004

[19] J.A. Gogeuen, C. Linde, Techniques For Requirements Elicitation, First International Symposium on Requirements Engineering, IEEE Computer Society Press, pp: 154-164, 1993.

[20] J.A. Goguen, Social issues in Requirements, http://www.cs.ucsd.edu/goguen/pps/sissues.ps, 5/6/2004.

[21] J.A. Goguen, Formality and Informality in Requirements Engineering, ICRE, pp:102-109, 1996.

[22] Goodrich, Victoria, and Olfman, Lorne, An Experimental Evaluation of Task and Methodology Variables for Requirements Definition Phase Success, Proceedings of the 23rd Annual Hawaii International Conference on System Sciences, 201-209. IEEE Computer Society, January 1990.

23] M. Harsu, A survey on domain engineering, http://practise.cs.tut.fi/pub/papers/domeng.pdf, 3/10/2004.

[24] A.M. Hickey, A.M. Davis, Elicitation technique selection: How Do Experts Do It?, Proceedings of the 11th IEEE International Requirements Engineering Conference, 2003.

[25] A.M. Hickey, A.M. Davis, Requirements elicitation and Elicitation Techniques Selection, Proceedings of the 36th Hawaii International Conference on System Science, 2003.

[26] Jokela, Timo, Lindberg, Kai, Statecharts Based Requirements Analysis: Deriving User Oriented Models, Microprocessing and Mi-croprogramming, vol 30, No 1-5:289-296, August 1990.

[27] S.B. Kuchmistaya, Incorporation of Joint Application Design (JAD) requirement Determination, Information Systems Analysis (MSIS 488), 2001.

[28] A.V. Lamsweerde, Requirements Engineering in Year 00: A Research Perspective, De;partment d'Inge;nierie Informatique, Universite; de Louvain, B-1348 Louvain-la-Neuve , 2000.

[29] P.D. Leedy, J.E. Ormrod, Practical Research, Seventh Edition, pp: 147-171, 2001

[30] J.C.S.P. Leite, Franco, A Strategy for Conceptual Model Acquisition, First IEEE International Symposium on Requirements Engineering, SanDiego, Ca, IEEE Computer Society Press, pp 243-246, 1993.

[31] N. Maiden, G. Rugg, ACRE: Selecting Methods for Requirements Acquisition, Software Engineering J., 11, pp. 183-192, 996.

[32] Management Decision and Research Center (MDRC), Using Qualitative Research Methods, http://www.mdrc.research.med.va.gov/mgmt_research_in_va/methodology/qualitative_research.cfm#4, 5/31/2004.

[33] W. L. Manning, Privacy and Confidentiality in Clinical Data Management Systems: Why You Should Guard the Safe, Medical Records, Privacy & Confidentiality, Summer 1995.

[34] D. Maurer, T. Warfel, Card Sorting: a definitive guide, http://www.boxesandarrows.com.archives/card_sorting_a_definitive_guide.php, April 2004, 4/19/2004.

[35] T. Menzies, S. Easterbrook, B. Nuseibeh, Sam Waugh , An Empirical Investigation of Multiple Viewpoint Reasoning in Requirements Engineering, IEEE International Symposium on Requirements Engineering, June 1999.

[36] H. Mouratidis, A Novel Agent-Based System to Support the single Assessment Process of older people, journal of Health informatics, pp 149-162, September 2003.

[37] B. Nuseibeh, S. Easterbrook, Requirements Engineering: a roadmap, Proceeding of the conference on The future of Software Engineering, pp.35-46, 2000.

[38] *D. Richards*, A Process Model for Requirements Elicitation, In Proceeding of the 11th Australasian Conference on Information Systems, December 2000.

[39] L.G. Robert, "Searching for the Holy Grail of Software Engineering," Communications of the ACM, Vol. 45, No.5, 2002.

[40] S. Robertson, Requirements Trawling: Techniques for Discovering Requirements, International Journal of Human-Computer Studies, 55, , pp. 405-421, October 2001.

[41] G. Rugg, Elicitation overview, http://mcs.open.ac.uk/gr768/elicitation/overview/qm/1.shtml, May 2003, 4/9/2004

[42] G. Rugg, Laddering, http://mcs.open.ac.uk/gr768/elicitation/index.shtml, May 2003, 2/20/2004.

[43] E.H. Shortliffe, L.E. Perreault, The Computer Meets Medicine and Biology: Emergence of a Dicipline, Informatics-Review , Vol. 2, No. 15 , August 1999.

[44] Standish Group, The Chaos Report, http://instruct.uwo.ca/engin-sc/se351a/readings/Standish-Group-Chaos-94.pdf, 1994, Nov.27,2003

[45] Standish Group, The Chaos Report, http://www.Standishgroup.com/Sample-research /PDFPages/Chaos1998.pdf, 1998, Nov.27, 2003.

[46] Turban, Knowledge Acquisition, Expert Systems and Applied Artificial Intelligence, pp: 117-139, September 1993.

[47] Zahniser, A. Richard, How to Speed Development with Group Ses-sions, IEEE Software, pp109-110, May 1990.

[48] Zucconi, Lin, Techniques and Experiences Capturing Requirements for Several Real-Time Applications, ACM SIGSOFT Software Engineering Notes 14(6):51-55, October 1989.

Appendix A

The following table shows the applicability of each of the techniques to the health-care domain. The values are given based on the study made in *section 4.3*

Technique	Applicability
Apprenticing	1
Card sorting	1
Delphi Techniques	1
Discourse Analysis	1
Document Analysis	2
Ethnography	2
Focus Group	1
IBIS	2
Interaction Analysis	2
Introspection	1
JAD session	1
Laddering	2
Mind mapping	1
Modeling	2
Observation	2
Protocol Analysis	0
Prototyping	2
Questionnaires	0
Requirement Reuse	2
Role playing	1
Scenario Analysis	2
Structured interview	1
Team building	1
Unstructured Interview	1
Video and Audio	1
War gamming	1

Table 8: Applicability of techniques to the health-care domain

Appendix B

I. Traditional Techniques

Techniques under this category involve those techniques that are usually used to collect any kind of information [37]. The main characteristic of most conventional approaches is that they assume preexisting categories. They also consider the system as static throughout the course of the elicitation activity. In this document, three of them are discussed: *interviews*, *questionnaires*, and *document analysis techniques*. Each of technique has its own set of assumptions that it took for granted, as best application conditions.

a. Interviews

Interview is one of the most commonly used questioning techniques across different domains [19]. Interview can be used to elicit tacit knowledge as well as hard facts, opinions, feelings, and goals. It also help to identify work flows, factors that influence the operation of systems and elements that make up the system like documents, procedures and policies. Interview needs considerable preparation on the part of the interviewer [14]. Before conducting interviews the analyst should read background material, establish interview objective, whom to interview and etc.

The integration and decision making process after using interview is time consuming and difficult. Since requirements are volatile, the longer this process takes the more the needs of stakeholders deviate from the collected requirements.

Interviews are used for new projects in a new domain, whenever heavy politics are there, and when the user and customer are inaccessible. The success of interview is determined by

the knowledge of the interviewer in the problem domain, solution domain and analyst's skill of communication.

There are two commonly known interview types, which are structured interview and unstructured interview. Unstructured interview is a kind of informal interview. It needs little planning and is used basically as starting point to quickly understand the structure of the problem domain. Unstructured interview is used to get high level concepts by allowing the subject to give idea freely. The interviewer poses questions and makes the subject to respond freely i.e. the interviewer doesn't force the subject to follow specific terms. Although unstructured interviews are less constrained, they are still limited by the need for participant to share basic concepts and methods, so they can negotiate shared meanings.

Structured interviews are more directed and constrained, requiring more planning and used to gather detail information. Since structured interviews are goal oriented and systematic we often use them to look for an answer to a pre-defined set of questions. They are also useful when the interviewer is familiar with the problem domain and the categorization that exists in the client organization. Thus, often, structured interviews follow unstructured interviews.

On the other hand, interviews gives much room for bias; they need domain knowledge and often when people are asked they respond what must be done instead of what they actually do. These are some of the disadvantages of interview technique. Moreover, in interview, a question may not be answered and if we have too many respondents, arranging the interview session and analyzing the response would be time taking.

Interviews can work with other techniques in order to produce better results. For example it can be used with elicitation techniques such as discourse analysis, interaction analysis.

b. Questionnaires

Questionnaire is very widely used technique. Questionnaire can be used when the size of the stakeholders is large enough, and the issues addressed are clear enough to all concerned. The validity of questionnaires depends on the sample size and the selection of the target audience. Questionnaires fail when subjects are asked about topics that they don't have ways to talk about, or do not want to talk about or do not know.

Questionnaires assume that a given question has the same meaning to all subjects [21]. But in reality categories and concepts that are transparent to one community can be entirely opaque to members of another community.

Thus, if questionnaire is going to be used then it must be ensured that the subject's category is clearly identified and there exists concrete baseline understanding of the problem and the solution domain. Questionnaire also needs enough amount of preparation time before conducting the distribution of the questionnaire.

c. Document analysis

Document analysis is a technique for extracting requirements from existing documents. The documents can be official policies, work procedures, manuals, magazines, pamphlets, and memos.

Document analysis can be used to get mission and goals of a company, duties and responsibilities of individuals, sections and divisions as well as, detail work procedures of activities. The output of document analysis can be used to structure the different elicitation techniques and used as a reference baseline for the validity of other techniques. Completeness of requirement and conflicts in requirements can be detected with the help of document analysis.

In order to get started in understanding terminologies, the analyst may need someone who could introduce the available documents to the requirements engineer. A data model is a useful tool for recording the results of document analysis and raising questions about the underlying requirements [40].

II. Group techniques

Group techniques are techniques, which involves interactive group work and support the documentation process. In this category *JAD/RAD*, *focus group*, *team building*, *Issue-Based Information System (IBIS)* and *Delphi Technique* techniques will be discussed.

Group techniques should be used when there is general belief that "missed opportunities" exist and when a large, diverse, and autonomous set of stakeholders exists. It is used to envisioning future system, breaking the constraints of current system knowledge that would dominate interview or observation [24]. Group techniques are also important to collect data about existing system form end users. But the drivers for group techniques are multiple stakeholders, disparate needs, and a demand to reach consensus soon and when there is difficulty to proceed with the current status without agreeing on some important issues. Group techniques are not profitable when we have geographically dispersed stakeholders in the absence of appropriate communication technology, or when stakeholders are very busy and it is difficult to arrange meetings or when there is no enough time to prepare for the group sessions.

Group techniques are used to explore team dynamics to get richer understanding of the needs of stakeholders. Group techniques require skills like communication, leadership, ability to facilitate meetings, modeling, and conflict resolution and creativity [24]. Often the group composition includes developers, users, customers, modulator (facilitator) and

recorder (scribe). The success of group techniques depends on a type of the involved participants and on the skill of the modulator.

The purpose of group meetings can be different of different elicitation sessions. It can be to solve problems, classify issues, brainstorm solutions to problems, resolute conflicts, conduct reviews, collect and merge facts and data, report progress, and assign tasks.

In the following sections we will see some of the group techniques in some detail:

a. Joint Application Development (JAD)

JAD is a technique, using group sessions, ensuring that information is gathered from all affected parties, and that requirements are received in outcome are approved by all participants, and not only by decision of the system analysts collecting the requirements [27]. In other words JAD facilitate the development of the system requirements with the agreement of the analysts, customers, users and other stakeholders that would be affected by the development of the new system. JAD's main theme is to bring together representatives from different groups, which potentially influence the development and usage of the system into a structured workshop to foster timely decision-making. JAD is part of Rapid Application Development (RAD) tool kit, and is a series of highly structured group interview sessions aimed at reaching consensus on a project's goals and scope [18].

The make-up of JAD team is crucial to the success of JAD sessions. Thus, the right representatives should be selected. JAD technique may be costly because it involves many people for elongated duration of time. Arranging meetings with this requirement will not be simple especially if individuals involved in the JAD team are busy.

Even if JAD needs enough preparation, it minimizes the time with requirements elicitation process because agreed requirements would be generated fast. It also brings experts together and creates a sense of ownership. This is especially important when the client company is ignorant of computer-based applications.

If enough pre-workshop preparation is not done before a JAD session, the outcome can be dangerous. It could negatively affect members of the team and existing healthy system causing for uncomfortable and even hostile environment. The other problem with JAD technique is the problem that comes because of the status difference of members in the organization. This may cause some not to say what they really feel and want to say, especially if it is something they think is not familiar to others. Use of some techniques like prototyping, modeling and IBIS techniques improves the outcome of JAD sessions [4].

b. Focus Group

Focus group assembles experts together and discusses the problem. The discussion can be structured like debate or unstructured such as simple discussion on an issue. Focus group is a kind of group interview [19], it is widely use in marketing research, when new products are going to be introduced. Focus group allows more natural interaction among members of the group.

Even though group facilitators do not impose their category on the group, there is a possibility that members impose their category on other members during focus group meeting. It is also often difficult for non-technical members of focus groups to understand some technical concepts and stay active in the group.

One of the problem with focus groups is the composition of the group i.e. the members are not from natural communities. The other problem is focus group doesn't help in eliciting design requirements because the subjects are not software experts.

c. Team building

Effective elicitation requires teamwork [24]. Team building is a second order elicitation technique i.e. it doesn't provide direct requirement output rather it enhance effectiveness of other elicitation techniques. This technique is applicable in situations where stakeholders have no previous group work experience and when there is a strong feel of opinion difference among them. Family therapy brings stakeholders to act like a family because they are around a common problem.

d. Issue-Based Information System (IBIS)

In the requirements elicitation process, documenting rationales is one of the most important activities especially in organizations where there is high staff turnover. It is also important to have documented rationales of requirements when there is high rate of requirements change. IBIS is a structured technique, which allows the rationale underplaying requirements to be organized and tracked [7]. It is used to capture detail dialogue information by keeping track of the issues being discussed, the positions on the issues, and the arguments in support of or objecting to positions.

IBIS is easy to learn, can be supported by tools, and is non-intrusive technique i.e. it doesn't affect communication among stakeholders. IBIS also made group meetings more productive by helping group members to focus on single issue and when side issues arise putting them for other sessions, when important. It also helps focus thinking on complex, difficult, or critical parts of the problem, and improves detection of incomplete or

inconsistent thinking. IBIS also force assumptions and definitions to be made explicit early in the development process.

IBIS doesn't help in the selection of best positions, has no support for ideas outside of issues, positions and arguments under process. IBIS has no automatic consistency checking mechanisms because the output of IBIS is not formalized.

e. Delphi Technique

Delphi technique is used in a condition where contact between experts is difficult. In this technique, each expert submits judgments regarding requirements and then all judgments are circulated anonymously to all experts. Each expert then submits a revised judgment. This will be done iteratively until judgments converge.

III. Simulation Techniques

The best way to get requirements, which are not easily available, is to stimulate the subject to recall requirements by providing stimulants. In this category, techniques like scenario *analysis, prototyping, models, role-playing,* and *war-gaming* techniques will be discussed.

a. Scenario analysis

Scenarios are stories, which explain how a system might be used. Discovering scenarios exposes possible system interactions and reveals system facilities, which may be required. Scenario analysis technique produces natural language output and is good for requirements procurement.

Since scenarios are inherently partial; they raise a coverage problem similar to test cases making it impossible to verify absence error [35]. The detail description of scenarios force premature choice of design option and over specification. Scenarios are too much labor intensive to collect and document them more over there are no recommendations on how

scenario-based requirements engineering should be practiced, even less tool support is available [2]. Scenarios are good models for interactive systems and for systems that corners the end user as a central subject of the project.

Since scenarios have low cost and limited expressiveness they are most appropriate for communicating specific system features in situations of high uncertainty. They are also good for eliciting requirements that could be forgotten by subjects. Scenarios also facilitate communication among stakeholders and anchors requirements analysis in real world experience.

b. Prototyping

A prototype software system is one that simulates the important interfaces and performs the main functions of the intended system [3]. Unless for some special purpose, prototype shouldn't necessarily be bound by most non-functional requirements such as hardware, speed, size or cost constraints put on the final product. Prototypes do not also worry to handle exceptions that need to be handled latter. The purpose of prototype is to make real the conceptual structure specified, so that clients can test for consistency and usability and developers to check technical feasibility.

Commonly there are two types of prototyping, which are *throwaway prototyping and evolutionary prototyping.* Throwaway prototyping is used to see the different components of the target system; they will not be used in the development of the final product. Evolutionary prototyping is used for well-defined requirements and the prototype will develop to final product.

Prototype allows users to experiment and discover what they really need to support their work [3]. Developers will also be able to check feasibility before high development cost is

incurred. Prototypes are used to demonstrate the 'look and feel' of the user interface and used for systems that need serious consideration of user interfaces. They can also be used to identify requirements, which are hidden, forgotten or neglected, and that can cause incomplete, ambiguous or conflicting requirements.

It is not advisable to use prototyping when it is known that it would take extended time and in the absence of mutual trust between the developing company and the client company [24]. It is good technique when there is a great deal of uncertainty about requirements, or where early feedback from stakeholders is needed [19, 37]. Prototyping can also be used to describe requirements related to decision support systems and to simulate critical systems.

Training cost, development cost, and extended development schedule are some of the constraints that hider the wide usage of prototypes in the software industry.

c. Models

Models like DFD, state chart, activity diagram, UML, etc plays an important role in the process of requirements elicitation [24]. They can visually show the static and dynamic nature of the system in a clear and unambiguous way. Stakeholders can easily see the conflicts and incompleteness of requirements from domain models.

Nowadays models are used to facilitate communication among stakeholders, uncover missing information, organize information gathered from other techniques, and uncover inconsistencies that exist in the captured requirements. Models also show the problem raised by the user and the corresponding solution.

Difficulties to be understood by the users, inclusion of too many design constraints, and pre-decision of development approaches are some of the problems of the conceptual models [11].

ER models or class diagrams are good choices, if the system is data-centric application. Models are also good choices when using system archeology (document analysis) to summarize what is found from documents [23]. Models technique is also good candidate when the user interface is very important for the final product [26].

d. Role-playing

This technique is used when key stakeholders are inaccessible. It uses surrogates, where non-stakeholders play the role of stakeholders [24]. This technique can be used for example if the user or the customer is unreachable, people from out side will be selected to act in place of them. Even if it is not in natural setup, it will help to formalize the process of the development in the absence of the client.

e. War-gaming

War gamming is used when we have time and one of the following happens. Situation is emotional, there exists a risk of large loss, the problem is ill defined or we don't know the actors or events.

This technique can be used for example when developing sensitive systems like national terrorists information hijacking system.

IV. Cognitive Techniques

Cognitive techniques focus on the individual perception of the problem and the solution for the problem. In this category *protocol analysis*, *card sorting*, *laddering* and *Introspection* techniques will be discussed.

a. Protocol analysis

Protocol analysis asks a subject to engage in some task and concurrently talk aloud, explaining the subject's thought process [46]. The expert will be asked by the analyst to think out aloud while carrying out the task [46]. Hence, we can say that protocol analysis is 'do and talk aloud', explaining thought process, technique [19]. The assumption of the protocol analysis technique is that people can produce language that gives a trace of autonomous cognitive activity and thus protocols do represent and are related to actual cognitive process [15].

The major disadvantage of using this technique is that the technique doesn't consider any context, from where the requirement is elicited. Language is a communication tool that involves at least two individuals but in the setup of protocol analysis, the subject will talk aloud, the analyst will just track what the subject is talking. It doesn't deal with requirements in their natural setting. Therefore, we can say that it is more of artificial setup.

Protocol analysis can be used when it is possible to use the techniques with little setup, when there is age, background, etc similarity between the analyst and the subject, when there is no task procedure manuals, when task is complex and the subject is unable to describe it without doing the task, or when the subject is busy and the requirements engineer has little knowledge on the problem domain [19]. Thus, protocol analysis is good to elicit semi-tact knowledge when the problem is taken for granted or working memory.

b. Card sorting

Card sorting is a cheap, simple, amenable to automation elicitation technique that is used for classification of knowledge. Card sorting process involves sorting a series of cards, each labeled with a piece of content or functionality, into groups that make sense to users or

participants [34]. For a given set of domain objects, written on cards experts are asked to sort the cards into groups then says what the criterion was for sorting, and what the groups were. Card sorting enables you to know how people categorize items and concepts [17].

Card sorting is a reliable method for finding patterns in how users would expect to find content or functionality [34]. The output of card sorting can be used for latter interface design and for establishing model for hierarchical relationships. Card sorting is user-centered method for increasing a system's usability. Card sorting is effective once you have completed identification of confusing terms, functional requirements of users or the content of a system. Card sorting is good technique when we have readily available packages and when respondents are familiar with the packages [31]. Card sorting is also good choice when identifying data requirements of a system. Moreover, it is used to identify items that are likely to be difficult to categorize and terminology that is likely to be misunderstood [17].

Card sorting doesn't consider user's task, results may vary, analysis can be time consuming, and may capture only 'surface' characteristics of the system to be developed. Before implementing the technique suitable entities need to be identified with suitable semantic spread across domain. The other problem is that card sorting is not scalable technique (i.e., as the number of cards increases the variation among respondents will increase and that make the analysis difficult).

c. Laddering techniques

Laddering is a structured technique that uses a set of probes (type of questions) to acquire structure and content of stakeholders' knowledge [41]. Originally laddering was used for knowledge acquisition for knowledge-base systems [37]. Laddering deals with hierarchical

knowledge and interview can be used to move up and down a conceptual hierarchy. Often card sorting follow laddering technique [31].

Laddering is good for selecting package or setting criteria for the right package when there is less knowledge of packages [25, 31]. It is also used to identify goals, values and explanations of subjective terms. Recently, it is used for eliciting explanations and clarifications of technical and subjective terms. Laddering is used to identify the data requirement of a system. It is also used to elicit subclasses when going up the hierarchy and explanations when going down the hierarchy [42].

The major problem of laddering is its assumption that knowledge is always hierarchical. Hence, laddering is not advisable for eliciting knowledge, which is not hierarchical.

d. Introspection

It is imagining 'what kind of system I want if I were doing the job, using this equipment and in this environment. The introspection of an expert in a different field is unlikely to reflect the experiences of the actual user [21]. Therefore, this technique is good when the analyst is the problem domain expert. Introspection without careful consideration of its limits can be highly inaccurate.

Introspection is limited by the expert's imagination. It is difficult to introspect what work settings look like, or the conditions under which a new technology will be learned or used.

V. Content Analysis

There are many textual, video and audio products of other techniques such as interview, observation, video-based technique, scenario and etc. In this category, we will see techniques, which could be used to analyze those products to elicit more requirements, to

clarify existing requirements and to resolute conflicts. The techniques, which will be presented, involve: *mind mapping, video-based elicitation, Requirement Reuse,* and *content analysis*.

a. Mind mapping

Mind mapping is a way of taking more extensive and meaningful notes. With a mind map you use words, pictures, symbols and colors to capture the way that your brain perceives the subject matter [40]. Mind mapping is used for summarizing data collected during interview or any other techniques. For example, it can be used to enrich our notes after an interview session. Mind mapping is also good candidate to get undreamed requirements.

b. Video and audio technique

The idea of this technique is to derive new requirements, especially those related to usability, by evaluating individuals' facial expressions, words of body language, and their wording while they are using the software or are working [40]. Video-based elicitation can be used to identify informal practices in the workplace [21].

Video and audio technique may not be used everywhere due to legal and ethical issues. It will also be costly when the size of the subjects is very large and dispersed geographically. The analysis cost of data collected using this technique will not also be simple. It can take a highly skilled person a very long time to produce a transcript from a videotape of live interaction.

c. Requirement Reuse

Reuse involves taking the requirements, which have been developed for one system and using them in a new similar system. The basic goal of requirements reuse is to be able to determine that the needed system's feature or part of its functionality have already been

stated by another system's requirements, which with small alterations could effectively be deployed in a new context.

Requirements reuse opens the door for analysis, design, code, test and operational procedures reuse. The main output of software ruse is quality and productivity.

Requirements reuse includes selection of the relevant artifact and synthesis of the new requirement activities.

Requirements reuse can produce more benefit than only design or code reuse because traceability relationships can be established between the high level requirements of the system and the low level implementation, which are built from them.

VI. Immersion techniques

In all the previous categories the analyst is outside the system and tries to measure some parameters that could show the requirements of the stakeholders. But when we come to this category the analyst will be immersed in the users' system and look and compile the moment-by-moment activity of the clients. In this category we will see *ethnography, interaction analysis, apprenticing, observation techniques*, and *discourse*. Discourse and interaction analysis are only applicable to situations where there is significant social interaction; conversation and discourse analyses are only applicable to verbal data. But the most important limitation of these methods is that they are very labor intensive. Another limitation is that these methods cannot be (directly) applied to the study of systems that have not yet been built. However, they can be used to obtain tacit knowledge, because they bypass the unreliable explanations of users, and instead examine what they actually do.

a. Ethnography

Ethnography is designed to study the group in its natural setting for a lengthy period of time, often several months or even several years [29]. The focus of investigation is on the every day behaviors (e.g., interactions, language, rituals) of people in the group, with intent to identify cultural norms, beliefs, social structures, and other cultural patterns. Ethnographies were first used in cultural anthropology, but they are now seen in sociology, psychology, and education as well. Ethnography is especially useful for gaining an understanding of the complexities of a particular, intact culture. If ethnography is used the analyst should keep detailed notes of all work practices, analyze them and draw conclusion from them.

In the course of using ethnography several methods can be deployed in the process of the requirements identification such as interviews, observation, modeling, prototyping and etc. Ethnographic methods are followed by verbal methods, like interviews acquiring knowledge to generate scenarios [31]. Ethnography assumes spending time with people establish a trust relationship.

Unlike interviews and questionnaires ethnography uses the members own categories. Ethnography technique is used to elicit the actual work processes, which is often different from formal, prescribed process. Ethnography can be used when users are too busy and there is enough time for the elicitation process. Through immersion in data from some particular social group (such as stock broker), particular competencies are gradually acquired that let an analyst be a sensitive, effective ‘measuring instrument’ in that domain [21]. Thus, the best way to measure something is to be part of the measurement tool.

There are two major disadvantages of ethnography. The first and most challenging is that it needs long time to implement and the other is ethnography is labor intensive i.e. projects may involve hundreds of hours for recording transcribing and analyzing data.

b. Discourse analysis

It is used to elicit tacit knowledge. It uses two types of subanalysis to study the structure, which are interactional (comes from ethnomethodology) and linguistic (comes from sociolinguistics) [19].

Discourse analysis can follow other techniques such as interviews. Critical discourse analysis is grounded in interactional talk and text data, which employs sociolinguistic methods [1].

c. Interaction analysis

Interaction analysis is used to elicit tacit knowledge (i.e., knowledge which can not be easily accessed). Observing actual interactions in the work place is the best way when subjects are unable to articulate their work [19]. This technique is applicable when there is a feel of incompleteness or conflicts in the requirements. It is a technique, which considers the context in which the requirement resides.

d. Apprenticing

Apprenticing uses the idea of a master craftsman and an apprentice. The apprentice observes what the master does, asks questions and then tries to learn the work by doing some of it. This technique is usable when there is no schedule constraint and the job is easier to learn [40].

e. Observation technique

Sometimes, the best way to get the right understanding of what people actually doing is to observe them while they are in their workplace. An analyst may make observations either as an outsider or, in the case of ethnography, as a participant observer [41]. The primary advantage of conducting observation, in spite of other techniques that follow formalized and more structured approach, is its flexibility i.e. the analyst can easily shift focus as new data come to light [29]. Other techniques are more directed and formalized Observer spends time with the subjects, joining in, long enough to become a member of the group.

Observation is a good technique to elicit semi-tacit and tacit knowledge [31]. It is also used to reveal detail knowledge about the target requirement that other methods cannot elicit.

Appendix C

For the probes from the project environment a kind of decision matrix is used, where the analyst can choose from the given options. For each probe 5 options, 0 to 4, are given and the meaning of each value for each of the probes is different. The interpretation of each of the options against the probes is given below*:*

No	Probe	Option	
1	The purpose of the project is to maintain existing system	0	Minor modification to some features of an existing system
		1	Some features modification
		2	New feature addition to an existing system
		3	Major change to an existing system
		4	Totally new system
2	The target system is an interactive system	0	Not interactive at all
		1	Fairly interactive (i.e. few interfaces are there)
		2	Moderately interactive
		3	Highly interactive
		4	Extremely interactive
3	Relationship between analysts and clients is good	0	Hostile relationship
		1	Passive relationship
		2	Good relationship
		3	Very good relationship
		4	Hand and glove relationship
4	Practices in the client organization are documented	0	No documentation at all
		1	Only establishment documents are available
		2	Major policies and responsibilities are available
		3	Policies, work procedures, job descriptions are available
		4	Almost everything in the client organization is documented
5	Key stakeholders are easily available	0	They are not available at all
		1	They are available once in a blue moon
		2	They are fairly available
		3	They are most of the time available
		4	They are available all the time
6	Users have no trouble articulating their need	0	They are totally unable to verbalize their needs
		1	They can articulate some of their needs
		2	They are good in presenting their needs
		3	They are very good in verbalizing their needs
		4	They are extremely capable of stating their needs clearly

No	Probe	Option	
7	Users are computer literate	0	Users are not totally exposed to computers
		1	Users have simple utilities level experience
		2	Users are exposed to basic computer applications
		3	Users have good experience in using computer applications
		4	Users have extensive experience on using different applications
8	Analysts are familiar with the problem domain	0	No exposure at all
		1	They are fairly familiar
		2	They are moderately familiar
		3	They are highly familiar
		4	They are extremely familiar
9	The project is under serious schedule constraint	0	There is no schedule constraint at all
		1	It is under light schedule constraint
		2	It is under moderate schedule constraint
		3	It is under tight schedule constraint
		4	It is under extremely tight schedule constraint
10	The project is under serious financial constraint	0	There is no financial constraint at all
		1	It is under little financial constraint
		2	It is under moderate financial constraint
		3	It is under serious financial constraint
		4	It is under extremely serious schedule constraint
11	There is enormous and constant flux of stakeholders	0	Stakeholders are stable
		1	There is little instability of stakeholders
		2	Stakeholders are moderately unstable
		3	Stakeholders are highly unstable
		4	Stakeholders are extremely unstable
12	Stakeholders highly diversified	0	Stakeholders are totally uniform
		1	Stakeholders are fairly uniform
		2	Stakeholders are moderately uniform
		3	Stakeholders are diversified
		4	Stakeholders are very diversified
13	Unhealthy competitive spirit exists among stakeholders	0	Stakeholders have family relationship
		1	There is little competitive spirit which is not clearly visible
		2	There is a competitive spirit that sometimes appears and seen clearly
		3	There is a competitive spirit that is often clearly seen
		4	Stakeholders are against each other

No	Probe	Option	
14	Communication technology is available for remote stakeholders	0	There is no communication technology except physically going there
		1	There is only postal communication facility
		2	There is postal and telephone communication technology
		3	There is postal, telephone and e-mail communication technology
		4	There is postal, telephone, e-mail and group meeting technology
15	Reusable requirements are available	0	There is no reusable requirements asset
		1	There are some reusable requirements that are not well documented
		2	There are reusable requirements that are documented
		3	There are lots of reusable requirements that are well documented
		4	There are enormous reusable requirements which are automated for use
16	Stakeholders are large in number	0	They are few in number
		1	They are fairly large
		2	They are large in number
		3	They are very large in number
		4	They are extremely large in number

The applicability matrix of each of the techniques to the situation of the project environment is shown below.

Environmental Situations	Option	Document Analysis	Questionnaires	Structured interview	Interview JAD session	Protocol Analysis	Video and Audio	Scenario Analysis	Observation	Ethnography	Discourse Analysis		Introspection	Card sorting	Laddering	Focus Group	IBIS	Modeling	Apprenticing	Team building	Requirement Reuse	Mind mapping	Delphi Techniques	War gamming	Role playing	Interaction Analysis
The purpose of the project is to maintain existing system	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	1	1	1	1	1	1	2	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	2	1	1	1	1	2	1	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	3	1	1	2	2	2	1	1	1	1	1	1	1	1	1	2	2	1	1	1	1	1	2	1	1	1
	4	1	1	2	2	2	1	1	1	1	1	1	1	1	1	2	2	1	1	1	1	1	2	1	1	1
The target system is an interactive system	0	1	1	1	1	1	0	1	0	1	1	1	0	1	0	1	1	1	0	1	1	1	1	1	0	1	1
	1	1	1	1	1	1	0	1	0	1	1	1	0	1	0	1	1	1	0	1	1	1	1	1	0	1	1
	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	3	1	1	1	1	1	2	1	2	1	1	1	2	1	2	1	1	1	2	1	1	1	1	1	2	1	1
	4	1	1	1	1	1	2	1	2	1	1	1	2	1	2	1	1	1	2	1	1	1	1	1	2	1	1
Relationship between analysts and clients is good	0	2	0	0	0	0	0	1	0	1	1	2	1	1	0	1	1	1	1	0	2	2	1	1	1	2	1
	1	2	0	0	0	0	0	1	0	1	1	2	1	1	0	1	1	1	1	0	2	2	1	1	1	2	1
	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	3	1	1	2	2	2	2	1	1	1	1	1	1	1	1	1	1	2	1	2	0	1	1	1	1	1	1
	4	1	1	2	2	2	2	1	1	1	1	1	1	1	1	1	1	2	1	2	0	1	1	1	1	1	1

Environmental Situations	Option	Document Analysis	Questionnaires	Structured interview	Unstructured Interview	JAD session	Protocol Analysis	Video and Audio	Scenario Analysis	Observation	Ethnography	Discourse Analysis	Prototyping	Introspection	Card sorting	Laddering	Focus Group	IBIS	Modeling	Apprenticing	Team building	Requirement Reuse	Mind mapping	Delphi Techniques	War gamming	Role playing	Interaction Analysis
Practices in the client organization are documented	0	0	1	1	1	1	2	1	1	2	2	1	1	1	1	1	2	2	1	1	1	1	1	1	1	1	1
	1	1	1	1	1	1	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	3	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	4	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Key stakeholders are easily available	0	1	2	0	0	0	1	1	1	2	2	1	1	2	1	1	0	0	1	1	0	1	1	1	1	2	1
	1	1	2	0	0	0	1	1	1	2	2	1	1	2	1	1	0	0	1	1	0	1	1	1	1	2	1
	2	1	1	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1
	3	1	1	1	1	2	1	1	1	1	1	1	1	1	1	1	2	2	1	1	2	1	1	0	1	0	1
	4	1	1	1	1	2	1	1	1	1	1	1	1	1	1	1	2	2	1	1	2	1	1	0	1	0	1
Users have no trouble articulating their need	0	2	1	0	0	0	2	2	0	2	2	2	2	2	2	2	0	0	2	2	1	1	2	1	1	1	2
	1	2	1	0	0	0	2	2	0	2	2	2	2	2	2	2	0	0	2	2	1	1	2	1	1	1	2
	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	3	1	1	2	2	2	1	1	1	1	1	1	1	1	1	1	2	2	1	1	1	1	1	1	1	1	1
	4	1	1	2	2	2	1	1	1	1	1	1	1	1	1	1	2	2	1	1	1	1	1	1	1	1	1

Environmental Situations	Option	Document Analysis	Questionnaires	Structured interview	Unstructured Interview	JAD session	Protocol Analysis	Video and Audio	Scenario Analysis	Observation	Ethnography	Discourse Analysis	Prototyping	Introspection	Card sorting	Laddering	Focus Group	IBIS	Modeling	Apprenticing	Team building	Requirement Reuse	Mind mapping	Delphi Techniques	War gamming	Role playing	Interaction Analysis
Users are computer literate	0	1	1	1	1	2	1	1	2	1	1	1	2	1	2	1	2	2	2	1	2	1	1	1	1	1	1
	1	1	1	1	1	2	1	1	2	1	1	1	2	1	2	1	2	2	2	1	2	1	1	1	1	1	1
	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Analysts are familiar with the problem domain	0	2	0	0	1	1	1	1	1	2	2	1	1	0	1	2	1	1	1	2	1	2	1	1	1	1	1
	1	2	0	0	1	1	1	1	1	2	2	1	1	0	1	2	1	1	1	2	1	2	1	1	1	1	1
	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	3	1	1	2	2	1	1	1	1	1	1	1	1	2	1	1	1	1	1	1	1	1	1	1	1	1	1
	4	1	1	2	2	1	1	1	1	1	1	1	1	2	1	1	1	1	1	1	1	1	1	1	1	1	1
The project is under serious schedule constraint	0	1	1	1	1	1	1	1	1	2	2	2	2	0	1	1	2	2	1	1	1	1	1	2	1	1	1
	1	1	1	1	1	1	1	1	1	2	2	2	2	0	1	1	2	2	1	1	1	1	1	2	1	1	1
	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	3	1	1	1	1	2	1	0	1	0	0	0	0	2	1	1	1	1	1	0	0	2	0	0	0	2	0
	4	1	0	1	1	1	1	0	1	1	0	0	0	2	1	1	1	1	1	0	0	2	0	0	0	2	0

Environmental Situations	Option	Document Analysis	Questionnaires	Structured interview	Unstructured Interview	JAD session	Protocol Analysis	Video and Audio	Scenario Analysis	Observation	Ethnography	Discourse Analysis	Prototyping	Introspection	Card sorting	Laddering	Focus Group	IBIS	Modeling	Apprenticing	Team building	Requirement Reuse	Mind mapping	Delphi Techniques	War gamming	Role playing	Interaction Analysis
The project is under serious financial constraint	0	1	1	1	1	1	1	1	1	1	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	3	1	1	1	1	1	1	0	1	1	0	1	1	2	1	1	1	1	1	0	1	2	1	1	0	2	1
	4	1	1	1	1	1	1	0	1	1	0	1	1	2	1	1	1	1	1	0	1	2	1	1	0	2	1
There is enormous and constant flux of stakeholders	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	3	1	1	1	1	2	1	1	1	0	2	1	1	1	1	1	2	2	1	1	1	1	1	1	1	1	1
	4	1	1	1	1	2	1	1	1	0	2	1	1	1	1	1	2	2	1	1	1	1	1	1	1	1	1
Stakeholders are highly diversified	0	1	0	2	2	1	1	1	1	1	1	1	1	2	1	1	1	1	1	1	1	1	1	1	1	1	1
	1	1	0	2	2	1	1	1	1	1	1	1	1	2	1	1	1	1	1	1	1	1	1	1	1	1	1
	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	3	1	1	1	1	2	1	1	1	2	2	1	1	1	2	1	2	2	1	1	2	1	1	2	2	1	1
	4	1	1	1	1	2	1	1	1	2	2	1	1	1	2	1	2	2	1	1	2	1	1	2	2	1	1

Environmental Situations	Option	Document Analysis	Questionnaires	Structured interview	Unstructured Interview	JAD session	Protocol Analysis	Video and Audio	Scenario Analysis	Observation	Ethnography	Discourse Analysis	Prototyping	Introspection	Card sorting	Laddering	Focus Group	IBIS	Modeling	Apprenticing	Team building	Requirement Reuse	Mind mapping	Delphi Techniques	War gamming	Role playing	Interaction Analysis
Unhealthy competitive spirit exists among stakeholders	0	1	2	2	2	2	1	1	1	1	1	1	1	1	1	1	2	2	1	1	0	1	1	2	1	1	1
	1	1	2	2	2	2	1	1	1	1	1	1	1	1	1	1	2	2	1	1	0	1	1	2	1	1	1
	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	3	2	1	0	0	0	1	2	1	2	2	1	1	1	1	1	0	0	1	1	2	1	1	1	1	1	1
	4	2	0	0	0	0	1	2	1	2	2	1	1	1	1	1	0	0	1	1	2	1	1	1	1	1	1
Communication technology is available for remote stakeholders	0	2	1	1	1	0	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1	1	1	1	1	1	1
	1	2	1	1	1	0	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1	1	1	2	1	1	1
	2	2	2	2	2	0	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1	1	1	1	1	1	1
	3	2	2	2	2	0	1	1	1	1	1	1	1	1	1	1	2	2	1	1	1	2	1	2	1	2	1
	4	2	2	2	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	1	0	1	2	1
Reusable requirements are available	0	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1
	1	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1
	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2	1	1	1	1
	4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2	1	1	1	1

Environmental Situations	Option	Document Analysis	Questionnaires	Structured interview	Unstructured Interview	JAD session	Protocol Analysis	Video and Audio	Scenario Analysis	Observation	Ethnography	Discourse Analysis	Prototyping	Introspection	Card sorting	Laddering	Focus Group	IBIS	Modeling	Apprenticing	Team building	Requirement Reuse	Mind mapping	Delphi Techniques	War gamming	Role playing	Interaction Analysis
Stakeholders are large in number	0	1	0	2	2	2	2	2	1	2	1	1	1	1	2	1	2	2	1	1	2	1	1	2	1	1	1
	1	1	0	2	2	2	2	2	1	2	1	1	1	1	2	1	2	2	1	1	2	1	1	2	1	1	1
	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	3	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	4	2	2	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Appendix D

The following table contains applicability of techniques to goals:

Technique	g001	g002	g003	g004	g005	g006	
Apprenticing	0	0	1	2	1	1	0
Card sorting	0	1	1	1	1	0	0
Delphi Techniques	0	1	1	1	1	1	2
Discourse Analysis	1	1	1	2	2	1	1
Document Analysis	2	2	2	1	2	1	1
Ethnography	2	2	2	2	2	1	1
Focus Group	1	2	2	1	2	2	2
IBIS	1	1	1	2	2	2	2
Interaction Analysis	1	1	1	1	2	2	1
Introspection	1	1	2	1	0	0	0
JAD session	1	2	2	2	2	2	2
Laddering	2	1	1	2	2	2	0
Mind mapping	1	1	1	1	1	2	1
Modeling	0	2	2	1	1	2	2
Observation	2	2	2	2	2	2	1
Protocol Analysis	0	0	1	2	0	2	1
Prototyping	0	2	2	1	1	2	1
Questionnaires	1	1	1	1	1	1	1
Requirement Reuse	2	1	2	1	1	1	1
Role playing	0	0	1	0	1	1	0
Scenario Analysis	0	1	1	2	1	2	1
Structured interview	1	1	1	2	2	2	2
Team building	1	0	1	1	1	1	2
Unstructured Interview	2	2	2	1	1	1	1
Video and Audio	1	1	1	2	1	2	1
War gamming	0	2	1	0	1	2	1

g001=Identification of organizational context, g002=Identification of boundaries of the target system, g003=Identification of features of a system, g004=Detailed investigation of a given feature, g005=Identification of rationales for requirements, g006=Clarification of uncertainties in requirements, g007=Conflict resolution

Table 9: Applicability matrix to goals

DECLARATION

The thesis is my original work, has not been presented for a degree in any other university and that all sources of material used for the thesis have been duly acknowledged.

Semahegn Abebe Dr. Yirsaw Ayalew (Advisor)